A Taxonomy for Denial-of-Service Attacks in Wireless Sensor Networks

Anthony D. Wood and John A. Stankovic
Department of Computer Science
University of Virginia
{wood,stankovic}@cs.virginia.edu

Abstract

*Wireless sensor networks (WSNs) are transitioning to real-world applications, where they face attacks already experienced by the Internet and wireless ad hoc networks. One such attack is that of denial-of-service (*DoS*), which we believe will only become more prevalent as sensor networks become more pervasive and accessible. With the inherent resource limitations of WSN devices, they are particularly susceptible to the consumption and destruction of these scarce resources. We present a* DoS *attack taxonomy to identify the attacker, his capabilities, the target of the attack, vulnerabilities used, and the end result. We survey vulnerabilities in WSNs and give possible defenses. Protecting WSNs against* DoS *attacks—while remaining low-cost and flexible—is a primary research challenge that bears further exploration.*

1 Introduction

Interest in wireless sensor networks (WSNs) continues to build momentum, with research results and technology beginning to transition to real-world applications. The application possibilities include sensing and actuating in many types of environments, from monitoring remote environmental sites or hostile battlefields to controlling the modern comfort of indoor health-care facilities [ASSC02, YOD+02, KKP00]. Some networks may be used in places unsafe or undesirable for humans; others may be unobtrusively integrated into dense urban environments.

The WSN design space centers around small, wireless devices with magnetic, acoustic, optical, chemical or other sensors on-board [HSW+00]. Each is limited in resources, so nodes must communicate and coordinate to enact aggregate behaviors. Sensing nodes may not be serviceable once deployed, so low-power operation is essential for network longevity. Low per-unit cost and effective distributed algorithms will enable deployment at a large scale, where individual sensors are no longer identifiable or important. Failures are masked by robust aggregation, redundancy, and adaptive reconfiguration.

All the promises of robustness, fault-tolerance, and cost-effective operation hinge on overcoming the inherent resource limitations of sensor nodes. Memory and processing cycles are limited, as in many embedded devices. Since the energy-budget is small, wasted cycles and especially wasted radio transmissions are not wise. Without diligent consideration of these constraints, WSN protocols and architectures may never make it from the research lab to the field.

Security is another important, and frequently overlooked, requirement in many of these systems. Sensitive data generated by the network should be protected from unauthorized disclosure. Data and control mechanisms likewise should provide integrity and authenticity guarantees. These are all the more critical when the network itself can impact the environment using actuators or automatic responses.

Even with confidentiality and integrity, the WSN is not achieving its objectives if the services provided by it are not available to authorized users when they need it. In networks with such scarce resources, their improper consumption or destruction is a big concern. In addition to being a security problem, an inability of the network to perform its task may be a safety hazard, depending on the system being monitored or controlled.

This paper explores denial-of-service vulnerabilities in WSNs. We begin by examining terminology, the definition of denial-of-service, and why it is a potential problem for WSNs. In section 2 we present a taxonomy for classifying attacks. We describe a key part of the

taxonomy—vulnerabilities—in section 3, along with possible defenses. We conclude with an overview of related work in denial-of-service research and some final comments.

1.1 A Note on Terminology

Some of the terms used in this paper vary in their popular definition, depending on the author. For clarity and consistency, we adapt the following definitions from the National Information Systems Security Glossary [Com03]:

attack: Attempt to gain unauthorized access to a service, resource, or information, or the attempt to compromise integrity, availability, or confidentiality. Note that success is not necessary.

attacker, intruder, adversary: We use these terms synonymously to mean the originator of an attack.

vulnerability, flaw: Weakness in system security design, implementation, configuration or limitations that could be exploited.

threat: Any circumstance or event (such as the existence of an attacker and vulnerabilities) with the potential to adversely impact a system through a security breach.

risk: Probability that an attacker will exploit a particular vulnerability, causing harm to a system asset.

We continue with a more detailed look at denial-of-service itself.

1.2 Denial-of-Service

Classically, the definition of denial-of-service (DoS) involves three components: *authorized users*, a *shared service*, and a *maximum waiting time* [Gli83]. Authorized users are said to deny service to other authorized users when they prevent access to or use of a shared service for longer than some maximum waiting time. Though this definition appears straightforward, its application raises several important questions.

Who are the users?

Certainly, the users of a WSN as a whole are humans. Traditionally, processes acting on behalf of those humans are also considered users of the system. In a distributed general-purpose computing network, this is likely the end of the matter.

In a WSN, however, general interactive computing facilities are not likely to exist in any of the devices. The large number of devices, their relative inaccessibility, and low energy supplies are more manageable when localized autonomy and coordination are present. In such an environment, in-network services (such as localization, routing, and power management) are of more direct benefit to the sensor nodes that interact with them than to the human operators. Rather than treat only the human deployers, owners, or monitors of the network as users, it may be more useful to consider individual sensor nodes as users with respect to in-network services.

Distinctions such as *insider* versus *outsider* threats can then apply at two levels. At the highest level, insiders are those who have an employee–employer or other trusted relationship with the WSN owners. On a lower level, individual, authentic sensor nodes are insiders. If their identities or secrets are stolen, say by physical tampering, an intruder is afforded the access, opportunity and credentials to appear as an insider to the network. This possibility is affirmed in [And99], which suggests that "software agent/mobile code technology may represent a non-human variant of the 'insider threat.' "

A particular difficulty on the Internet is that the network is vast and open, such that it is currently not possible to authenticate every user. For some services, then, there are effectively no unauthorized users, since users are not strongly authenticated at all.

A WSN is a more closed and controlled system; but even here authentication is not without difficulty. Large-scale networks, and those in which periodic replenishment is expected after initial deployment, pose a variety of trust and identity management issues. We do not explore them further here.

What are the shared services?

Depending on who the users are, the WSN may provide a few aggregation services, such as monitoring or control of an area. Shared access to these types of services could take the form of multiple battlefield commanders querying a WSN for the location of chemical or biological hazards.

In-network services, such as routing, localization, and time synchronization, are also used by multiple sensor nodes concurrently. For this reason, it may be useful to view individual sensor devices as users, as discussed above.

How long is too long to wait?

Many systems may only have loosely specified deadlines for service fulfillment. Strictly interpreting the DoS definition to preclude the possibility of DoS attacks is not

useful in practice. Even discretionary services such as the World Wide Web have limits imposed by cognitive perceptions of action and response.

Passive WSNs may be able to tolerate widely varying service fulfillment times, as long as the waits are "reasonable," perhaps probabilistically. Networks that can control their environment are more likely to require real-time properties consistent with precise specification of maximum waiting times.

Finally, we note that modification or corruption of the intended and advertised service may also be considered denial-of-service, as in [Gli83]. In this case it is insufficient to deliver the *appearance* of service within some bounded time; only the *specified* service will do. Detection of this type of DoS is not as simple as setting a timer, and may be impossible. Though modification of the service is a DoS to its legitimate users, it is perhaps more appropriately the concern of ensuring the accuracy and correctness of system function.

We choose a broader definition, one inclusive of the above possibilities, adapted from the National Information Assurance Glossary [Com03]:

denial-of-service: The result of any action that prevents any part of a WSN from functioning correctly or in a timely manner.

Even if a more precise definition is elusive, we observe that a DoS attack usually has the following properties:

- **Malicious**—The act is performed intentionally, not accidentally. Accidental failures are the domain of fault-tolerance and reliability engineering. Since such failures can potentially produce equally disruptive results as DoS attacks, these fields have important contributions to make to the robustness of WSNs. They are not considered DoS, however, due to the lack of malice.

- **Disruptive**—A "successful" DoS attack degrades or disrupts some capability or service in the WSN. If the effect is not measurable, for example if it is prevented altogether, we may still say that an attack has occurred, but DoS has not. Note that disrupting the affected service may not be the end goal of the attacker.

- **Asymmetric**—Often the effect of an attack is much greater than the effort required to mount it. For example, sending a forged packet that overflows a remote buffer takes little effort, but may crash the server until an operator intervenes. Even in distributed-denial-of-service (DDoS) attacks, the effort to "recruit" zombies and issue an order to flood a victim is small compared to the flood of traffic that reaches the target. This kind of asymmetry is not necessary, but makes an attack easier and more economical for the perpetrator.

- **Remote**—Especially in distributed systems, an attacker usually can (and wishes to) carry out an attack over the network. Often this is by unauthenticated or lightly authenticated users (such as those possessing a valid return IP address, in the case of TCP-SYN floods [SKK+97]). The high profile of many types of DoS attacks would make physical presence uncomfortable for the attacker.

With this understanding of what a DoS is, we now explore its relation to other fields of study.

1.3 Relation to Other Fields

Denial-of-service is impacted by other fields, such as security, reliability, performance, and software engineering. In some cases, we can directly transfer advances in these areas to gains in DoS resistance.

Security

As described above, denial-of-service is a breach of the security characteristic of *availability*. Along with availability, *confidentiality* and *integrity* are the primary concerns of security. Integrity failures can themselves cause DoS; for example, clients are denied service if the server's replies are always corrupted. However, ensuring integrity is not sufficient to prevent DoS.

Protection models generally express what users *may* access, not that they will be able to regardless of what other users do [Gli83]. This is the domain of availability and fairness.

Reliability

Results from the areas of reliability and fault-tolerance may help a WSN be more robust to DoS attack. Loss-concealing and self-correcting coding schemes and protocols may increase the work an attacker must do to cause DoS. Reliability considers faults to be random and independent, however, in contrast to the intentional failures induced by a DoS attack.

Performance

Many DoS attacks consist of simple flooding, that is, overwhelming link bandwidth with large or numerous

messages. Since large commercial sites must process many requests or messages in a timely fashion as part of routine business, they are designed with fast, efficient network stacks that minimize interrupts and data copying. High-performance systems, such as web servers, will therefore naturally tolerate larger-scale DoS attacks based on flooding [KMW01]. Although the resource-impoverished sensor devices in WSNs are in many ways the opposite of these high-performance server farms, some of the efficiency techniques used may be applicable. Of course, efficiency is not sufficient to thwart every DoS attack, even those which involve flooding.

Software Engineering

Denial-of-service made possible by implementation flaws in programs continues to be underrepresented in the literature [And93]. This is despite the fact that exploit programs are frequently published on the Web that allow intrusion or DoS of vulnerable computers on the Internet. According to NIST's ICAT vulnerability search engine, there were 335 remotely exploitable availability-impacting vulnerabilities published in the year 2002 [ICA].

The large number of implementation-related flaws highlights the continuing need for research in software engineering as it can impact the process of writing secure programs.

1.4 Sensor Network Vulnerability

Though WSNs can be a low-cost, low-overhead way to monitor an environment in real-time, some of their attributes make them even more susceptible to DoS attack or damage. The attribute that dominates all others has already been discussed: *limited resources*. A device with scarce resources is at risk of resource consumption under normal circumstances. When an adversary is actively attempting to improperly consume or destroy its resources, the situation is worse. Other attributes are:

- **Remote location**—Networks that are distant or unmonitored have a greater response time if manual (physical) intervention is required.
- **Large scale**—Due to the large number of devices likely to be deployed, manual intervention on each device is not feasible within cost constraints. If a vulnerability is discovered and exploited in the sensor program code, it will be no small matter to physically collect, reprogram, and redeploy each device. Wireless code download is a possible remedy, but is not without its own security concerns.
- **Cost sensitive applications**—For a large-scale deployment to be cost effective, the per-unit cost of sensor devices must be low. This applies added pressure to control hardware and software development costs. In the real world, this translates to hasty designs and numerous implementation errors.
- **Application specificity**—Resource constraints may dictate that well-defined and uncoupled network layers are compressed or merged, reducing code modularity. Unforeseen interactions between network layers and services may give rise to new vulnerabilities.
- **Attractive target**—The systems monitored or controlled by the WSN may be safety-critical or highly visible, with significant consequences for failure. Depending on the motivation of the attacker, this may be precisely the profile of an attractive target.
- **Uncontrolled access**—Ubiquitous, wide-scale, and replenishable deployments may require relatively unfettered physical access to nodes. Odds of casual tampering or vandalism increase.
- **Middleware services**—As services are distributed among all or most nodes of the network, every device is a potential target for attack.

Effective mitigation of these inherent vulnerabilities requires careful consideration of the design and implementation of all protocols and code. To achieve a better understanding of the risks faced by the network, we now describe a taxonomy of DoS attacks against WSNs.

2 Attack Taxonomy

A taxonomy allows us to reason about attacks at a level higher than a simple list of vulnerabilities. It provides a classification system that ideally suggests ways to mitigate attacks by prevention, detection, and recovery. It can aid risk management by identifying vulnerabilities and making attacker characteristics explicit. Ideally, its insights can predict future attacks by exposing unguarded areas.

Every DoS attack is perpetrated by someone. The attacker has an identity and a motive, and is able to do certain things in or to the WSN. An attack targets some service or layer, exploiting a vulnerability. An attack may be thwarted, or it may succeed with varying results.

Each of these elements is necessary to understand the whole process of a DoS attack. Therefore, we believe a

useful and intuitive taxonomy should answer the following questions:

1. Who is the attacker?
2. What is she capable of?
3. What is the target?
4. How is it attacked?
5. What are the results?

The attack taxonomy we present here comprises the answers to each question in turn. Taken together, the attacker, capability, target, vulnerability, and results describe a DoS attack against a WSN. A summary of the taxonomy is shown in Figure 1. We now describe each part in detail.

2.1 Attacker

It is valuable to know who the attacker is in so far as it implies what they are likely to do and how well they may be able to do it. The following list characterizes an attacker according to four dimensions: *motive*, *determination*, *knowledge*, and *resources*. It bears some similarity to that presented by Howard in [How97].

- **Passerby**—Motivated by spontaneity; not determined; very little knowledge; few resources.
- **Vandal**—Desires to inflict damage, perhaps visibility; moderately determined; little knowledge; few resources necessary.
- **Hacker**—Desires access, motivated by curiosity and interest; highly determined; highly knowledgeable; moderate resources.
- **Raider**—Driven by personal or organizational monetary and/or political gain; highly determined; moderately–highly knowledgeable; moderate resources.
- **Terrorist or Foreign Power**—Causes real-world damage by compromise of critical systems, motivated by enmity; very determined; highly knowledgeable; very well resourced with time, money, and man-power.

This list is not meant to be exhaustive, but rather to cover a common and interesting range of attacker attributes. Those given represent a range of possibilities in each of the dimensions. Motivations vary, but the objective, for our discussion, is always to cause a DoS. In general, the more determined, knowledgeable, and well-resourced the attacker, the more difficult is the defense.

An alternate dimension is the *insider* versus *outsider* distinction, and is somewhat orthogonal to the above list, as any of the attackers could be an insider or collude with one. As discussed before, the term "insider" is not without ambiguity. It could refer only to the owner, operators, controllers, or monitors of the WSN. It could also refer to all processes executed on behalf of these users by sensor devices.

If commands are not guaranteed to be authentic by the security architecture of the WSN, these processes may include execution of malicious code by legitimate or subverted nodes on behalf of an adversary. This is similar to a payroll clerk executing an email attachment: some aspects of the attack profile should be based on who executes the attachment, rather than on who sent it.

We see a "trust continuum" between insider and outsider, instead of a bifurcation. Trust is concretely manifested by such aspects as knowledge, opportunity, authorization, affiliation, and access. Any moderately-sized organization will have multiple levels between the extreme insider (owner) and outsider (competitor). Not everyone has the combination to the safe or the ability to change payroll.

As discussed in reference [And99], insiders may be further divided into permanent, part-time, temporary, contractor, developers, and others. Each has a distinct profile that can inform an analysis of the identity of an attacker.

2.2 Capability

Knowing what an attacker is capable of is important for efficiently defending a WSN. Realistically, all possible vulnerabilities cannot be eliminated within resource, cost, or usability constraints. System designs are targeted to address threats that are most likely to be seen. Determining this real-world risk and selecting mitigation strategies depends partially on an enumeration of attacker capabilities, which we present.

We consider capabilities in several dimensions, each of which is described in detail below: number of attackers, their coordination, technical capabilities, and area of influence.

2.2.1 Number of Attackers

Clearly attacks may be mounted by *one* or by *many* attackers. Systems or particular defense strategies are often

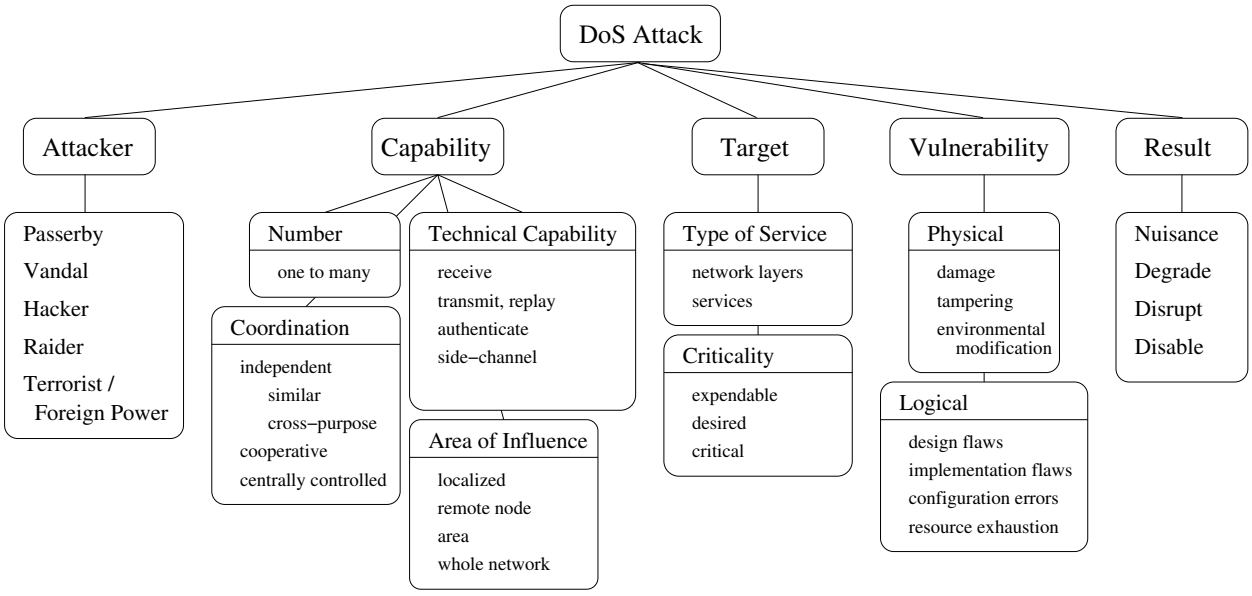

Figure 1: Taxonomy for denial-of-service attacks in wireless sensor networks.

classified by their resistance to compromise. They may tolerate one compromise per system, one per neighborhood, or one per time epoch. Some solutions that work against one attacker may fail if, for example, enough attackers are available to partition the network.

2.2.2 Coordination of Attackers

When multiple attackers are present, their coordination may vary. Each implies unique approaches to defense. The attackers may be *independent*, each attempting to cause DoS according to their individual motivations. If the attacks are *similar*, they may be aggregated and considered as N separate instances of the same attack. Or they may be working at *cross-purposes*, in which case there are N different attacks, possibily interfering with each other. For example, each may be attempting to selfishly gain priority in a routing service.

Autonomous attackers who *cooperate* to reach a common goal are generally more difficult to defend against. They may include a colluding authorized user (that is, an insider). A wormhole attack [HPJ03], in which two geographically separated nodes relay messages out-of-band and replay them (see subsection 3.8), is an example of this type of coordination.

Multiple attackers may also be *centrally controlled*, as in many Internet DDoS attacks. An adversary instructs many penetrated systems ("zombies"), sometimes through an indirect channel, to commence a DDoS attack against a victim. This kind of attack may be easiest to address by detecting and disrupting the control channel the attacker uses to disseminate instructions.

2.2.3 Technical Capability

Another aspect of an attacker model is the set of particular technical capabilities available. These capabilities are given here in increasing order of power or complexity.

An attacker may only be able to *receive* wireless radio transmissions. Limited to passive eavesdropping, the attacker cannot perform a DoS unless through other channels. For example, after homing in on a transmitting WSN node, the attacker may find and physically damage it. Listening to transmissions may precede other more complex attacks.

Along with receiving, if an attacker can *transmit* in the wireless channel used by the WSN, it can interact with sensor devices. It may be able to impersonate a legitimate node by replaying an old message, unless the network guarantees message freshness.

An attacker that can falsely *authenticate* itself to the WSN poses an even greater hazard. This may be possible through forgery or theft of legitimate credentials, or insider collusion. Depending on the authorization of the impersonated node, the attacker may have full access to all WSN services.

A *side-channel* may be available to the attacker for communication and coordination with other adversaries. Wired networks, other wireless channels, and optical communication allow the attacker to, for example, coordinate attacks despite disruption in the WSN's routing.

Finally, an attacker may have a *more powerful class* of devices than exist in the WSN. These include higher-bandwidth links, side-channels, superior computational facilities, wireline electric power, and mobility. It will be most difficult to overcome this kind of capability asymmetry.

2.2.4 Area of Influence

The area of an attacker's influence varies depending on the WSN and on the attacker's capabilities. An individual adversary possessing power and radio resources similar to a WSN node may only be able to affect a *localized* region. Nodes outside of this local communication range are unaffected.

Access to packet routing in the network gives the adversary the ability to affect *remote* nodes. Widely distributed base stations may be reachable from anywhere in the network.

Due to spatial properties of physical environments monitored by sensor networks, attackers may be able to affect an arbitrary *area* of the network. An area-multicast service may be abused to flood many nodes with each message sent by an adversary.

As a special case of the previous capability, an attacker may be able to affect the *whole* network, perhaps using an unauthenticated broadcast facility.

2.3 Target

Next we consider the attacker's target. Both the type of target and its importance to the network are factors that affect the overall risk and constrain solutions. To avoid speculation about the ultimate objective of the attacker, we define the target of a DoS attack to be the service that is being denied.

2.3.1 Type of Service

As described in [WS02], attacks may happen predominately against one layer or service in the network. They may also exploit service interactions. In a layered architecture, disrupting the lower layers is most advantageous for an attacker since most or all of the upper layers depend on it to function. Hence, an attack may affect multiple services by targeting their common underlying dependency.

Services include the typical layers found in a network stack: *physical*, *link-layer*, *network and routing*, *transport*, and *application*. Other WSN-specific services are also possible targets, such as: *localization*, *time synchronization*, *group management*, *directory services*, *entity tracking*, *power management*, *event detection*, *topography discovery*, *code download*, and *aggregation*. Any application-specific service may be targeted.

In a heterogenous network, attacks may also be targeted at particular devices. However, these attacks must take the form of an attempted exploitation of a vulnerability in some service provided by the node. Hence, we think it more useful even in this case to consider the type of service being attacked, rather than the specific identity of the victim.

2.3.2 Criticality of Target

The criticality of the target partially determines the risk associated with exploitation of related vulnerabilities. Critical services should be well protected against all forms of security violations, including DoS.

Some targeted services may be *expendable* or optional, such as the sensing coverage provided by a small number of nodes in a large WSN. The DoS at a few individuals may have no deleterious effect on the function of the network as a whole.

Other services may be *desired*, but non-critical. The network can achieve most or all of its purposes, though at a degraded level. Optimal function of the network is inhibited. For example, complete sensing coverage or equalized power consumption may be unattainable.

Critical services are those without which the WSN cannot function adequately. The loss of key services such as routing, directory services, or event detection may disrupt the entire operation of the network.

2.4 Vulnerability

After determining who an attacker is, what his capabilities are, and what the target is, we consider the vulnerabilities of the WSN. Vulnerabilities are weaknesses in the network, through which an attacker may gain or unduly exercise privilege. They arise due to conflicting requirements, cost constraints, designer shortcuts, and other reasons explored below. Where attacker ability and intention meet vulnerability, risk is created.

2.4.1 Physical

Most computer science literature considers only the damage that a malicious program or user can do using computer or network processes. Since our desire is to protect WSNs against all possible DoS attacks, we consider those perpetrated by low-tech physical means also.

Low-cost packaging is unlikely to be very resistant to intentional *physical damage*. Only physical presence is required to mount the attack—a low threshold for a network that may be deployed ubiquitously. Part or all of the WSN node may be damaged.

Though overt damage is easily detectable, more subtle *physical tampering* may go unnoticed. Here the node is physically modified in some way, perhaps to subvert it by replacing the program code. Or secret keys may be read from its memory and used in another attack.

Sensor networks function by being partially embedded in the real-world. This creates a unique vulnerability to *environmental modification* not found in traditional computer networks. An attacker who can systematically modify or falsify local sensor values in an area of the WSN may be able to mislead monitors or trigger improper control actions.

2.4.2 Logical

Logical vulnerabilities exist in computer programs and protocols. We identify four categories of these vulnerabilities.

Design flaws allow the use of a protocol that violates assumptions about the environment or manner of use, while conforming to the protocol specification. This may be caused by ambiguity in the specification, or by implementation variance among multi-vendor components. For example, a lack of authentication in a power management protocol may allow any node to put any other to sleep repeatedly.

Implementation flaws are errors in hardware construction or software coding. An example is inadequate boundary checking, which may result in a buffer overflowing with attacker-controlled contents. This may result in an access violation and crash, the execution of arbitrary code, or the destruction of a shared resource. Improper loop logic that allows a node to enter an infinite loop is another example of an implementation flaw that would cause DoS.

Configuration errors are the result of enabling improper settings for the particular environment or threat model appropriate to the WSN. Debug facilities in particular are often the source for such vulnerabilities.

Resource exhaustion is possible even if designs, implementations, and configurations are correct. An attacker that can generate large amounts of traffic can flood a victim's network link. Poorly authenticated memory allocation or code execution may also enable an attacker to consume these resources and cause a DoS.

2.5 Result

The final piece of the attack chain is the result of the DoS attempt.

An attack may only be a *nuisance* if the targeted service is not harmed, due to preventative mechanisms. Depending on service availability requirements, if the duration of the attack is short and the service quickly recovers, the DoS attack may not be successful. Nuisance attacks may result in the masking of other attacks, however.

Network performance may be *degraded* during the attack, but not stopped. Services may continue to function in the network as a whole, but are temporarily unavailable in certain regions of the network. The attack may be of short duration.

More serious, a service may be *disrupted* entirely for the duration of the attack plus some finite recovery time. The attack may last long enough to cause cascading failures in services dependent on the target.

A severe attack may result in a *disabled* service, in which the target ceases to function even after the attack stops. Such permanent disruption could be caused by, for example, physical damage, erased or corrupted memory, or crash failures without recovery. The attack or its consquences may last long enough to cause economic or physical damage.

3 Vulnerabilities and Defenses

Vulnerabilities are only one part of an attack, but it is useful to examine them in more detail. Without vulnerabilities the attack chain is broken and the WSN is perfectly secure (though not necessarily usable). Along with the available services that may be targeted, vulnerabilities are the only part of an attack under the control of the WSN designer.

Offering fewer services should be considered, if possible, since complexity is often a cause of insecurity. Reducing or streamlining services should reduce the opportunity for flaws to be created in design and implementation of the system. Smaller systems are easier to understand, and

make an exhaustive analysis of attack possibilities more tractable. For example, broadening the scope of an existing remote configuration facility may be safer, from a security standpoint, than adding a more general code-download facility.

System requirements dictate many of the services in a WSN, but designers should always minimize the risk from vulnerabilities. Depending on the purpose and environment of the WSN and the threat model assumed, some vulnerabilities may be tolerated. For example, a network deployed in the interior of a guarded and physically-secure military base may not require expensive tamper-proofing on every sensor device—unless the insider threat is significant.

Eliminating all vulnerabilities would prevent DoS attacks, but it is not possible in practice. Yu and Gligor argue in reference [YG90], that preventing DoS is only possible with the enforcement of agreements between users, external to the services being shared. While we therefore focus on detection and recovery from attacks, we also seek to limit DoS vulnerabilities wherever we can. Clearly some designs are more resilient in the face of attack than others. Prevention is most coveted, but when necessary we settle for detection and recovery [GBI00].

We assume that the attacker's objective, whatever his motivation, is to cause DoS against one or more of the services in the network (see section 2.3.1). He may use any available vulnerability or—more problematic—a combination of them to achieve this. Flaws in the system cannot be viewed only individually; the security risk depends on their aggregation. We intend to encourage this big-picture view with the DoS attack taxonomy already presented.

In the following sections, we describe some vulnerabilities of WSNs and possible defenses.

3.1 Jamming

Jamming is deliberate interference with radio reception to deny the target's use of a communication channel. For single-frequency networks, it is simple and effective, rendering the jammed node unable to communicate or coordinate with others in the network.

Constant transmission of a jamming signal is an expensive use of energy. If the attacker is limited in energy, as the WSN devices are, she may use sporadic or burst jamming instead. In this attack, she jams only when detecting radio transmissions in the area of the victim, which requires that she be nearby. Assuming that she can detect all such transmissions, and that she can react to them fast enough to jam before they finish, complete wireless channel denial is possible.

Defense

The most common defense against jamming attacks is the use of *spread-spectrum* communication [PSM82, And01]. In frequency hopping, a device transmits a signal on a frequency for a short period of time, changes to a different frequency and repeats. The transmitter and receiver must be coordinated. Direct-sequence spreads the signal over a wide band, using a pseudo-random bit stream. A receiver must know the spreading code to distinguish the signal from noise.

Frequency-hopping schemes are somewhat resistant to interference from an attacker who does not know the hopping sequence. However, the attacker may be able to jam a wide band of the spectrum, or even follow the hopping sequence by scanning for the next transmission and quickly tuning the transmitter. Direct-sequence spread-spectrum is more effective at defeating the jammer, though its processing-gain advantage may be defeated by a high-power wideband jamming signal [Stå00].

Due to the sychronization and cost requirements, low-cost sensor devices may be limited to using the more susceptible single-frequency communication.

If jamming cannot be prevented, it may instead be detected and *mapped* by surrounding nodes [WSS03, CG03]. A description of the region may then be reported back to network monitors, who can use conventional means to remove the attacker. In-network knowledge of the extent of the jammed region may also allow for automatic routing avoidance or mobile jammer tracking.

Jamming always occurs at the receiver, that is, it does not interfere with transmissions. A sensor device with important data may temporarily overcome localized jamming by sending a high-power transmission to an unaffected node [WS02]. This node can then relay the message on behalf of the jammed node. Such a scheme must be used sparingly, however, since a high-power transmission will prematurely drain the device's energy—resulting in a permanent DoS.

If alternate modes of communication are available, such as acoustic, infrared, or optical [ASSC02], a node may switch to one of these schemes when the radio is jammed. Of course, these other channels may be jammed as well by a determined attacker.

3.2 Tampering

Large-scale WSNs may be deployed in densely populated areas, where physical access to individual nodes is impossible to prevent. Even casual passersby may be able to damage, destroy, or *tamper* with sensor devices. Destruction of the node could cause gaps in sensor or communication coverage.

More well-equipped attackers can interrogate a device's memory, stealing its data or cryptographic keys. Its code can be replaced with a malicious program—one potentially undetectable to neighboring nodes. The capability profile of the subverted node is now that of a fully authorized insider.

Defense
Like jamming, an attack at this low level can render the entire sensor network useless. One intrinsic deterrent to tampering is the *physical distribution* of the network, that is, the geographic separation of individual nodes. Most low-cost attacks require at least brief physical presence near the targeted nodes. Protection measures outside the scope of the WSN may be sufficient to discourage tampering attacks. For example, upon the detection, tracking, and reporting of human intruders within the compass of the network, guards or operators could physically intervene.

An obvious, albeit difficult, defense against tampering is to construct sensor devices with *tamper-resistant packaging* [AK96]. This is generally an expensive and imperfect approach, and one that must be narrowly tailored to an accurate threat model. Defending against accidental damage or curious mishandling is much easier than thwarting a determined and well-funded attack in an electronics lab.

Another approach is to *prevent detection* of the nodes. Camouflaging the packaging, hiding the device, and using low-probability of intercept (LPI) radio techniques are among the possibilities. Again, these may all increase the cost and complexity of WSN design.

3.3 Collisions

Similar to jamming the physical radio channel, an attacker can willfully cause *collisions* or corruption at the link layer. By detecting and parsing radio transmissions near the victim, the attacker can disrupt key elements of packets, such as fields that contribute to checksums or the checksums themselves. With little effort or duration of transmission, the attacker may be able to cause the victim to discard a much longer packet—wasting both the channel access and the transmitter's energy. Such disruption may trigger backoffs in contention-control mechanisms that delay other messages.

Defense
Unfortunately, in wireless networks detection of a collision with one's own transmission is difficult. Standard collision avoidance mechanisms do not help either, since they are cooperative by nature. An attacker simply ignores the avoidance protocol, and transmits at the same time as the victim.

Error correcting codes can be used to provide some protection against corruption of message data. They are well-suited for covering random transmission errors, in which independent bits of a message may be flipped. An attacker, though, can always corrupt more data than the code can correct. The codes themselves also cost additional processing and transmission overhead, as the message contains greater redundancy.

3.4 Exhaustion and Interrogation

An attacker may be able to DoS a network by inducing repeated retransmission attempts. Even in the absence of high-rate traffic, if a node must continually retransmit due to collisions, as described above, eventually its energy may be *exhausted*. The attacker need only corrupt a small part of a much longer message, or perhaps jam an acknowledgement from a neighbor. Randomizing backoffs is not a defense, as the attacker can easily listen and wait for the next attempt. As long as he can react before the entire packet is transmitted, the cycle can continue.

At various layers in the network small messages (such as queries) may elicit much larger responses. For example, an attacker may be able to replay a broadcast initialization command, causing nodes throughout the network to perform localization or time synchronization procedures. Such unauthenticated or stale messages provide an easy avenue for traffic amplification. We call this repeated solicitation of energy-draining responses *interrogation*.

Defense
Services can require that requests be *authenticated*, otherwise refusing to answer with even a negative acknowledgement. This may cause confusion with legitimate clients, however, who receive no indication of whether the service provider has failed or is ignoring them. Omitting extraneous messages from protocols improves message efficiency, but may result in brittle state sychronization between the endpoints.

Another defense is to *rate-limit* response to even properly authenticated nodes. Excessive requests will be queued

or ignored without sending expensive radio transmissions. The rate must be high enough to provide sufficient bandwidth and timeliness for authorized users.

A node detecting an interrogation attack may also *notify upper layers* of the condition. This is similar to the congestion or delivery-failure feedback that some link-layers provide to the network layer. Rate limitations may be adapted based on the recent history of request traffic.

3.5 Selective Forwarding

WSNs usually depend on every node to take part in routing for its neighbors if it can provide a desirable forwarding path. Various *selective forwarding* attacks can exploit this dependence to cause DoS via routing.

A subverted sensor device can simply *neglect* to forward certain messages. A random dropping policy raises the local loss rates and may induce costly end-to-end recovery mechanisms. An attacker may also drop messages to or from certain victims, such as base stations or other servers.

At an extreme, a node could not only refuse to forward any packets, but could advertise a desirable path to its neighbors, creating a routing *blackhole*. Any messages passing nearby will be diverted to the adversary, where they are silently dropped. In addition to causing a DoS to the senders of the messages, neighbors of the adversary suffer from increased contention due to the above-normal levels of traffic.

Defense
Using *multiple disjoint routing* paths [GGSE01] and *diversity coding* [ACLGM93] can mitigate the effect of the attack. These defenses lessen the probability that a message will encounter an adversary along all routes to the destination. Diversity coding sends encoded messages along multiple paths, such that the originals can be reconstructed to conceal message loss, without the cost of full duplication.

To counter these defenses, an attacker must subvert additional nodes along the disjoint paths or choose an important source to move closer to, where jamming will be effective.

Nodes can also *monitor* their neighbors to gain probabilistic assurance that messages are being correctly forwarded. A node relays a message to its neighbor and then listens to the wireless channel, noting whether it overhears the neighbor's subsequent broadcast [MGLB00] of the same message. Although collisions, collusion, and asymmetric communication links limit the sender's ability to monitor every packet, the forwarding ratio can be used to inform a quality-rating mechanism. This mechanism is responsible for choosing a next-hop neighbor that has a high probability of properly forwarding subsequent messages.

Periodic end-to-end *probing* [CL97] can also alert a node to troublesome network paths, whether from congestion or malicious neglect. If an adversary can distinguish the probes from normal traffic, however, he can properly forward them so as not to arouse suspicion.

3.6 Misdirection

By forwarding messages along wrong paths, an attacker *misdirects* them, perhaps by advertising false routing updates. An attacker could DoS a particular sender by diverting only traffic originating from the victim node. A receiver could likewise be denied service if the attacker diverts traffic away from the node. This may be possible by rewriting the downstream path in routing algorithms which embed source-routes in each packet.

An attacker can also forge a source address when sending a request, so that the response will return to the victim. This could be done to confuse the victim or to flood it, if a service provides a mechanism for traffic amplification.

Defense
Routing updates should be *authenticated* to prevent malicious modification by untrusted adversaries [ZH99]. A freshness mechanism can protect against replay attacks, while cryptographic integrity checks protect against unauthorized modification of a message while in transit.

In WSNs that use a hierarchical structure for routing, *egress filtering* may be appropriate. Nodes which serve as collection points for subordinates' traffic may examine each message before forwarding it. Messages with source addresses that could not legitimately originate at lower levels of the hierarchy are discarded.

3.7 Sinkholes

Karlof and Wagner describe *sinkhole* attacks [KW03], in which an attempt is made to lure traffic from the sensor network to pass through an adversary. Low-cost routes may be erroneously flooded to lure the traffic, or a wormhole attack (see below) could be mounted to actually provide a low-cost route. In either case, the objective is for the attacker to be positioned such that other selective forwarding attacks, or merely eavesdropping, are easier to do.

Defense
One approach to avoiding sinkholes is to use routing algorithms that are *resistant* to arbitrary configurations, such as geographic forwarding [Fin87, KW03]. Since each node makes an independent forwarding decision based on the location of its neighbors, it is not as easy to attract routing to an attacker.

Communicating parties may also use *end-to-end verification* of advertised latency or quality to detect when a path may contain an unwarranted diversion [KW03]. Upon detecting a problem, nodes may attempt *systematic rerouting* to avoid the malicious node [SBD02].

3.8 Wormholes

In a *wormhole* attack, adversaries cooperate to provide a low-latency side-channel for communication [HPJ03]. For example, two attackers may possess a second radio for communicating over a higher-power, long-range link. Messages received at one attacker are relayed to the other using the side-channel, where they are transmitted as if only one-hop away from the original source.

This ability to understate ones distance from another node may cause neighboring nodes to favor the attacker for routing—another example of a sinkhole. As long as the side-channel exists, service may actually be enhanced, instead of denied. However, when the attacker moves or ceases to tunnel messages, the network may be left in an inconsistent state that requires re-initialization of some services to restore proper function.

Defense
As described for sinkholes, *geographic forwarding* is a tamper-resistant routing protocol (though see the Sybil attack, below). Each message is forwarded individually, choosing the next-hop node to be the neighbor closest to the ultimate destination. Such a scheme would not favor a wormhole in the network, though it may coincidentally use it.

Hu et al. describe a defense based on *packet leashes*, where the distance that a message may travel in a single-hop is limited [HPJ03]. Each message includes a timestamp and the location of the sender. The receiver compares these with its own location and time to determine if the maximum transmission range has been exceeded. The solution requires clock synchronization and accurate location verification, which may limit its applicability to WSNs.

3.9 Sybil Attack

Most protocols assume that nodes present a single unique identity. In a *Sybil attack*, an attacker presents multiple identities [Dou02]. Coupled with insecure location claims, this means an attacker can appear to be in multiple places at the same time. By creating fake identities of nodes located at the edge of communication range all around a victim, chances are high that the attacker will be chosen as the next-hop in geographic forwarding. The attack can also degrade any guarantees made by a multi-path routing scheme, making selective forwarding easy.

Defense
Since identity fraud is central to the Sybil attack, proper *authentication* is a key defense [KW03]. A trusted key server or base station may be used to authenticate nodes to each other and bootstrap a shared session key for encrypted communications, as in SPINS [PSW+01]. This requires that every node share a secret key with the key server. If a single network key is used, compromise of a any node in the WSN would defeat all authentication.

Another defense is *location verification*. Sastry et al. describe a simple protocol that uses the difference in time-of-flight of radio and sound waves to securely verify location claims [SSW03]. The combination of these two defenses, verifying identities and locations, would prevent Sybil-based DoS attacks.

3.10 Flooding

A *flooding* attack overwhelms a victim's limited resources, whether memory, processing cycles, or bandwidth. In a homogeneous network, a subverted node may have the same resource limitations as a victim, making an attack relatively expensive to mount. However, if an attacker possesses or subverts a more powerful device, such as a base station or laptop, the cost of the flooding attack relative to the result is much lower.

Poorly designed protocols can provide a stage for traffic amplification attacks, such as the Internet Smurf [Cen98] attack. Here an attacker forges the victim's address as the source of a widely and easily distributed request. All nodes receiving the request reply back to the victim, flooding its link. Hence, a few messages from an attacker result in a storm of responses to the victim.

Defense
To combat the consumption of memory resources at servers, Aura and Nikander describe principles for *stateless connection* management. This approach securely stores the server's state in all messages, requiring the

client to return it with every future response. The server need not store the state associated with the connection, which could lead to memory exhaustion if connections are not strongly authenticated (as is the case with TCP SYN-floods [SKK+97]).

Even without the overhead of ferrying server state in all messages, protocols can avoid allocating resources for unauthenticated traffic. The need for protocols that create traffic asymmetries, such as area-multicast, should be carefully weighed against their potential to allow traffic amplification attacks.

Another approach to curtailing flooding is to require the clients of services to commit significant resources before connections are established. *Client puzzles* are one such method, whereby servers dispense cryptographic puzzles that must be solved by brute-force before connection-related resources on the server are allocated [JB99, ANL01]. The difficulty of the puzzles is scalable, so the server can increase the requirements when it believes it is under attack. In a WSN, this could adversely affect the many legitimate sensor devices, each of which has limited resources to commit.

A final strategy is to provide a way to detect the source of the flooding using a *traceback* mechanism. Existing schemes are IP-based and are appropriate for the Internet's scale and structure [SWKA00, SPS+02, DFS01]. The goal, which could be ported to a WSN environment, is to allow even spoofed messages to be traced back to their actual source, using in-network auditing, periodic return-path messages, and other mechanisms.

3.11 HELLO Floods

Slightly different from conventional flooding, a *HELLO flood* is a single broadcast by a powerful adversary to many members of the WSN, announcing false neighbor status [KW03]. Many protocols use the exchange of HELLO messages to establish local neighborhood tables. The result of a HELLO flood is that every node thinks the attacker is within one-hop radio communication range.

If the attacker subsequently advertises low-cost routes, nodes will attempt to forward their messages to the attacker. Retransmission attempts to the absent attacker cause traffic congestion and confusion in the entire routing system.

Defense
Verifying the *bi-directionality* of local links before using them is effective if the attacker possesses the same reception capabilities as the sensor devices [KW03]. However, if the attacker can use a sensitive receiver, it can eventually convince nodes in the network of its legitimacy.

Authentication, as described for the Sybil defense, is also a possible solution. Nodes can use a trusted third-party to verify the authenticity of each of its neighbors before forwarding messages to them.

3.12 Algorithmic Complexity Attack

An *algorithmic complexity attack* [CW03] requires several features to be present to be successful. First, a service must use some algorithm and data structure with super-linear worse-case behavior, such as a hash table that inserts colliding elements into a hash chain. The input to the algorithm must be controllable by the attacker. The attacker must be able to compute values that will evoke worst-case performance and be able to deliver these values to the service. Once the data structure is primed, the attacker rapidly sends data that takes a long time to process due to the inefficiency of the algorithm. During this time, processor cycles at the victim are consumed in a DoS attack. In the hash table example, this happens when all the inserted values hash to the same bucket and are appended to the chain after examining every other element—a $O(n)$ operation.

Defense
Removing any of the above requirements will disrupt the attack. When possible, services should use algorithms with *efficient* worst-case performance. The service can also *remove determinism* (from the point of view of the attacker) from the algorithm execution. In the hash table example, this can mean using a keyed cryptographic hash function, so that the sequence of hash values computed by the program is unpredictable for the attacker. Another simple approach is to *limit the size of data structures* so that even in the worst-case, less efficient algorithms cannot consume too much processing time.

The smaller memory sizes and lower-rate communication patterns that would ordinarily mitigate this kind of attack may be counteracted by the relatively slow microprocessors that may exist in WSN devices.

4 Related Work

We briefly describe related work in two primary areas: the development of DoS as a security concern, and the classification of DoS attacks.

Gligor presents an early definition of denial-of-service, as already discussed in section 1.2, in terms of users, services, and maximum waiting times [Gli83]. He argues that current operating system protection mechanisms are insufficient to address DoS. Inter-user dependencies are identified as a critical element of DoS. Gligor claims that *any* dependency between users of a service that do not explicitly share objects in the service, is undesirable and must be eliminated. He gives examples of DoS in operating systems due to bad sharing policies, mechanisms, and the interactions between them.

Subsequent work [Gli84, Gli86] refines the necessary and sufficient conditions for DoS and discusses the guises of DoS in computer networks. Gligor classifies attacks into three categories. Service-oriented attacks are local, and are due to sharing and user dependencies, as in [Gli83]. Intruder-oriented attacks involve the deletion or delay of packets in a network (modification and replay are handled by integrity models). Finally, performance fluctuations are due to congestion and flow-control.

Gligor claims that most intruder-based attacks cannot be prevented, and emphasizes the need to detect and recover from attacks. This idea was further developed by Yu and Gligor, who use temporal logic to express sharing agreements between users [YG90]. A key claim is that user agreements are external constraints and cannot be internalized and enforced by the shared service. A full grasp of the implications of this was slow in coming [GBI00]: prevention of DoS is impossible except where it can be reduced to a liveness or safety problem.

Millen describes a trusted service called the "Denial-of-service Protection Base (DPB)" which enforces user agreements for resource allocation [Mil92]. He suggests that probabilistic waiting-time policies are possible, though none are given. The DPB must be tamperproof, not prevented from running, implement all dependent services, and be able to revoke resources. These limitations present challenges for application of the DPB.

Needham briefly describes application-layer attacks against a server, network, and client, including generating false alarms [Nee93, Nee94]. He argues that in some circumstances availability is more important than either confidentiality or authenticity. Needham suggests that the best protection against DoS is for a monitor service to depend on the same services that it is attempting to protect.

Meadows addresses the problem that DoS attacks on the Internet often occur before parties are authenticated [Mea99]. She presents a specification method for describing the incremental trust and capabilities that communicating parties have after performing each step of a protocol. Unlike authentication, DoS is a matter of degree and no protocol is fully immune.

More recent work has focused on aspects of DoS attacks unique to wireless ad hoc networks [GHW02] and wireless sensor networks [WS02]. In the former, Geng et al. propose using dynamic usage-based pricing to provide an economic disincentive to potential DoS attackers. In the latter, Wood and Stankovic survey attacks and defenses against a WSN and examine the weaknesses of two protocols that did not include security in their original designs.

Other recent works include discussions of DoS possibilities alongside other concerns, such as secure routing in WSNs [KW03].

Various authors have presented taxonomies of attacks or vulnerabilities tailored to particular environments.

Howard employs a systematic, process-based taxonomy to categorize computer and network attacks on the Internet, including DoS [How97, §6.4]. Rather than listing particular vulnerabilities, he focuses on the attack as a whole, from an operational perspective. According to this taxonomy, an *attacker* uses *tools* to gain unauthorized *access* (through exploiting vulnerabilities), causing certain *results* consistent with the attacker's ultimate *objectives*.

We choose to explain attacks based (in part) on the vulnerabilities exploited, rather than the tools used. It is our observation that the programs and scripts used to exploit vulnerabilities become more automatic, complex, and widely-applicable over time. We also consider peculiarities of WSNs not appropriate to the wired Internet.

Leiwo et al. classify Internet DoS attacks as tolerable or fatal [LAN00]. Attack methods are classified based on the allowable deviations from a protocol: none, sequence, syntax, and semantics. They suggest that server resources only be allocated after client authentication, and that the workload of clients should be higher than that of servers.

Moore et al. approach DoS attacks empirically, using *backscatter analysis* [MVS01]. They loosely classify attacks by examining the victims' responses sent to spoofed addresses that happened to reside in a large network they were monitoring. During their three-week period of study, they recorded 12,805 attacks on more than 5000 hosts.

Mirkovic and Reiher present a taxonomy of both distributed-DoS (DDoS) attacks and defenses [MMR02]. Many of the attack characterizations are specific to distributed, centrally coordinated attacks on IP networks. They differentiate between protocol and brute-force (flooding) attacks on systems, noting that modifying protocols to protect against protocol attacks pushes the attack into the brute-force category. One part

of the attack taxonomy, "degree of automation" is similar to Howard's tool-based approach.

Rice and Davis take a different approach to categorizing attacks, performing a *post-mortem* analysis of attack tools to build a genealogical attack tree [RD02]. This has the advantage of making relationships between vulnerabilities explicit, and may also suggest future attack possibilities.

Shields classifies the results of a network denial-of-service attack as a four-tuple: the protocol or layer used, effect on the limiting resource, resource being consumed or corrupted, and the location of the device attacked [Shi02].

Lough discusses and reviews prior work on the properties of good taxonomies, and presents VERDICT, a new taxonomy for computer security flaws [Lou01]. VERDICT attributes the causes of vulnerabilities to improper verification, exposure, randomness, or deallocation. Lough applies the taxonomy to wireless networks, but only briefly considers DoS.

5 Conclusion

Wireless sensor network research is transitioning to interesting real-world applications. As the networks become more pervasive and accessible, they will face some of the same problems with which wired Internet and wireless ad hoc networks already struggle.

One such problem is denial-of-service. DoS attacks are commonplace on the Internet, from background noise to front page headlines. WSNs, with all their inherent resource limitations, are particularly susceptible to the consumption or destruction of these scarce resources. Protecting the availability of sensor networks that monitor and control critical systems—while remaining low-cost and flexible—is a principle challenge that remains.

Every DoS attack originates from someone, uses various methods, and produces results. We present an attack taxonomy to facilitate higher-level reasoning and analysis of risk in WSNs. It describes aspects of the whole process of a DoS attack:

- the attacker's identity, including motivation and resources;
- what the attacker is capable of doing in the WSN;
- type and criticality of service(s) targeted by the attack;
- physical and logical vulnerabilities that may be exploited; and,
- the results of the attack, from nuisance to catastrophe.

Since vulnerabilities are among the few parts of an attack under the control of the WSN designer, we review a variety of them along with possible defenses. Many solutions given are not without cost; but they may be a necessary price for increased robustness against DoS attacks. We seek to prevent DoS where that is possible, and detect, tolerate, and recover from the rest.

6 Acknowledgements

This work was supported in part by the DARPA NEST program under grant F33615–01–C–1905 and by MURI award N00014–01–1–0576.

References

[ACLGM93] E. Ayanoglu, I. Chih-Lin, R. D. Gitlin, and J. E. Mazo. Diversity coding for self-healing and fault-tolerant communication networks. *IEEE Trans. Comm.*, COM-41:1677–1686, November 1993. 3.5

[AK96] Ross Anderson and Markus Kuhn. Tamper resistance — a cautionary note. In *Proceedings of the 2nd USENIX Workshop on Electronic Commerce*, pages 1–11, Oakland, California, November 1996. 3.2

[And93] Ross Anderson. Why cryptosystems fail. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 215–227, Fairfax, Virginia, November 1993. ACM Press. 1.3

[And99] Robert H. Anderson. Research and development initiatives focused on preventing, detecting, and responding to insider misuse of critical defense information systems, CF-151-OSD. In *Workshop co-sponsored by RAND's NSRD*, 1999. 1.2, 2.1

[And01] Ross Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley Computer Publishing, 2001. 3.1

[ANL01] Tuomas Aura, Pekka Nikander, and Jussipekka Leiwo. DOS-resistant authentication with client puzzles. *Lecture Notes in Computer Science*, 2133:170–177, 2001. 3.10

[ASSC02] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks (Amsterdam, Netherlands: 1999)*, 38(4):393–422, March 2002. 1, 3.1

[Cen98] CERT Coordination Center. Smurf ip denial-of-service attacks. Technical report, CERT Advisory CA-98:01, November 1998. 3.10

[CG03] K. Chintalapudi and R. Govindan. Localized edge detection in wireless sensor networks. In *Proceedings of the IEEE ICC Workshop on Sensor Network Protocols and Applications (SNPA)*, May 2003. 3.1

[CL97] Steven Cheung and Karl Levitt. Protecting routing infrastructures from denial of service using cooperative intrusion detection. In *New Security Paradigms Workshop, Cumbria, UK*, September 1997. 3.5

[Com03] Committee on National Security Systems (CNSS). *National Information Assurance Glossary, NSTISSI No. 4009*, May 2003. 1.1, 1.2

[CW03] Scott A. Crosby and Dan S. Wallach. Denial of service via algorithmic complexity attacks. In *Proceedings of USENIX Security 2003*, August 2003. 3.12

[DFS01] Drew Dean, Matt Franklin, and Adam Stubblefield. An algebraic approach to IP traceback. In *Proceedings of Network and Distributed Systems Security Symposium (NDSS)*, pages 3–12, February 2001. 3.10

[Dou02] John R. Douceur. The sybil attack. In *In IPTPS*, pages 251–260, 2002. 3.9

[Fin87] G. G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISI/RR-87-180, ISI, March 1987. 3.7

[GBI00] Virgil D. Gligor, Matt Blaze, and John Ioannidis. Denial of service - panel discussion, April 2000. Security Protocols Workshop. 3, 4

[GGSE01] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *Mobile Computing and Communications Review*, 4(5), October 2001. 3.5

[GHW02] Xianjun Geng, Yun Huang, and Andrew B. Whinston. Defending wireless infrastructure against the challenge of DDoS attacks. *Mobile Networks and Applications*, 7(3):213–223, 2002. 4

[Gli83] Virgil D. Gligor. A note on the denial-of-service problem. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 139–149, 1983. 1.2, 1.3, 4

[Gli84] Virgil D. Gligor. A note on denial-of-service in operating systems. *IEEE Transactions on Software Engineering*, SE-10(3):320–324, May 1984. 4

[Gli86] Virgil D. Gligor. On denial-of-service in computer networks. In *Proceedings of the International Conference on Data Engineering*, pages 608–617. IEEE, 1986. 4

[How97] John D. Howard. *An analysis of security incidents on the Internet 1989–1995*. PhD thesis, Department of Engineering and Public Policy, Carnegie Mellon University, April 1997. 2.1, 4

[HPJ03] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Packet leashes: A defense against wormhole attacks in wireless networks. In *Proceedings of IEEE Infocom 2003*, April 2003. 2.2.2, 3.8

[HSW+00] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David E. Culler, and Kristofer S. J. Pister. System architecture directions for networked sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104, 2000. 1

[ICA] ICAT metabase: A CVE based vulnerability database. NIST. http://icat.nist.gov/icat.cfm, visited August 2003. 1.3

[JB99] Ari Juels and John Brainard. Client puzzles: A cryptographic defense against connection depletion attacks. In *S. Kent, editor, Proceedings of NDSS '99 (Networks and Distributed Security Systems)*, pages 151–165, 1999. 3.10

[KKP00] J. Kahn, R. Katz, and K. Pister. Emerging challenges: Mobile networking for 'smart dust'. *J. Comm. Networks*, 2(3):188–196, September 2000. 1

[KMW01] Frank Kargl, Jorn Maier, and Michael Weber. Protecting web servers from distributed denial of service attacks. In *Proceedings of the 10th International WWW Conference*, pages 514–524, 2001. 1.3

[KW03] Chris Karlof and David Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *First IEEE International Workshop on Sensor Network Protocols and Applications*, 2003. 3.7, 3.9, 3.11, 4

[LAN00] Jussipekka Leiwo, Tuomas Aura, and Pekka Nikander. Towards network denial of service resistant protocols. In *Proc. of the 15th International Information Security Conference (IFIP/SEC 200)*, pages 301–310. Kluwer, August 2000. 4

[Lou01] Daniel L. Lough. *A Taxonomy of Computer Attacks with Applications to Wireless Networks*. PhD thesis, Virginia Tech, Computer Engineering Department, April 2001. 4

[Mea99] Catherine Meadows. A formal framework and evaluation method for network denial of service. In *Proceedings of The 12th Computer Security Foundations Workshop*, pages 4–13. IEEE Computer Society Press, June 1999. 4

[MGLB00] Sergio Marti, Thomas Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM-00)*, pages 255–265, N. Y., August 2000. ACM Press. 3.5

[Mil92] Jonathan K. Millen. A resource allocation model for denial of service. In *Proc. 1992 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 137–147. IEEE Computer Society, May 1992. 4

[MMR02] Jelena Mirkovic, Janice Martin, and Peter Reiher. A taxonomy of DDoS attacks and DDoS defense mechanisms. Technical Report CSD-TR-020018, Computer Science Department, University of California, Los Angeles, 2002. 4

[MVS01] D. Moore, G. Voelker, and S. Savage. Inferring Internet denial-of-service activity. In *Proceedings of the 10th USENIX Security Symposium*, pages 9–22. USENIX, August 2001. 4

[Nee93] Roger M. Needham. Denial of service. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 151–153. ACM Press, 1993. 4

[Nee94] Roger M. Needham. Denial of service: An example. *Communications of the ACM*, 37(11):42–46, 1994. 4

[PSM82] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein. Theory of spread spectrum communications—a tutorial. *IEEE Transactions on Communications*, 20(5):855–884, May 1982. 3.1

[PSW+01] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks. In *Proceedings of Seventh Annual International Conference on Mobile Computing and Networks MOBICOM 2001*, pages 189–199, July 2001. 3.9

[RD02] Greg Rice and James A. Davis. A genealogical approach to analyzing post-mortem denial of service attacks. In *Secure and Dependable System Forensics Workshop*, September 2002. 4

[SBD02] Jessica Staddon, Dirk Balfanz, and Glenn Durfee. Efficient tracing of failed nodes in sensor networks. In *Proceedings of the first ACM international workshop on Wireless sensor networks and applications (WSNA)*, pages 122–130. ACM Press, 2002. 3.7

[Shi02] Clay Shields. What do we mean by network denial of service? In *Proceedings of the 2002 IEEE Workshop on Information Assurance and Security*, June 2002. 4

[SKK+97] C. L. Schuba, I. V. Krsul, M. G. Kuhn, E. H. Spafford, A. Sundaram, and D. Zamboni. Analysis of a denial of service attack on TCP. In *Proceedings of the 1997 Conference on Security and Privacy (S&P-97)*, pages 208–223, Los Alamitos, May 1997. IEEE Press. 1.2, 3.10

[SPS+02] Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Beverly Schwartz, Stephen T. Kent, and W. Timothy Strayer. Single-packet ip traceback. *IEEE/ACM Transactions on Networking (ToN)*, 10(6):721–734, December 2002. 3.10

[SSW03] Naveen Sastry, Umesh Shankar, and David Wagner. Secure verification of location claims. In *ACM Workshop on Wireless Security (WiSe 2003)*, September 2003. 3.9

[Stå00] Mika Ståhlberg. Radio jamming attacks against two popular mobile networks, 2000. Helsinki University of Technology, Tik-110.501 Seminar on Network Security. 3.1

[SWKA00] Stefan Savage, David Wetherall, Anna R. Karlin, and Tom Anderson. Practical network support for IP traceback. In *Proceedings of ACM SIGCOMM*, pages 295–306, August 2000. 3.10

[WS02] Anthony D. Wood and John A. Stankovic. Denial of service in sensor networks. *IEEE Computer*, 35(10):54–62, October 2002. 2.3.1, 3.1, 4

[WSS03] Anthony D. Wood, John A. Stankovic, and Sang H. Son. JAM: A jammed-area mapping service for sensor networks. In *Proceedings of the Real-Time Systems Symposium, to appear*, December 2003. 3.1

[YG90] Che-Fn Yu and Virgil D. Gligor. A specification and verification method for preventing denial of service. *IEEE Transactions on Software Engineering*, 16(6):581–592, June 1990. 3, 4

[YOD+02] Xiping Yang, Keat G. Ong, William R. Dreschel, Kefeng Zeng, Casey S. Mungle, and Craig A. Grimes. Design of a wireless sensor network for long-term, in-situ monitoring of an aqueous environment. *Sensors*, 2:455–472, 2002. 1

[ZH99] Lidong Zhou and Zygmunt J. Haas. Securing ad hoc networks. *IEEE Network*, 13(6):24–30, 1999. 3.6