Global Information Assurance Certification Paper

GIAC Security Essentials Certification (GSEC) Practical Assignment
Version 1.4b, Option 2

Rapid Tactical Reconnaissance Techniques for Extremely Large-Scale, Dynamic Enterprise Networks

Jonathan Ham

June 29, 2003

Contents

1 Summary

The most commonly employed tools and techniques for network-based reconnaissance do not scale to extremely large and dynamic enterprises, especially where immediate results are required for tactical responses to rapidly emerging threats. However, the introduction of massive parallelization, along with other techniques to improve efficiency, can improve the effectiveness and timeliness of such efforts, even in such environments.

References for this work have been selected from the entire past decade, in order to demonstrate both the timeless nature of the basic approach to tactical recon, and the evolution of automated tools and approaches to the problem.

2 The Problem

The problem of routine vulnerability discovery for mid- to large-scale enterprises has become a much discussed subject, even having been addressed recently in this forum [6]. However, most of the commonly employed tools and techniques become cumbersome and best, and ineffective at worst, when approaching enterprises which are both extremely massive in scale, and either highly decentralized or dynamic (or both).

While some 10,000 hosts might be routinely monitored to an acceptable degree with Grime's approach, keeping tabs on a somewhat dynamic pool of 50,000 networked devices — spread throughout an IP address space of roughly 500,000 addresses — requires an amount of additional effort. In this case study we will be dealing with just such an enterprise.

2.1 Background

The environment for this case study involves eight topologically contiguous "Class B" address blocks, and two additional "Class C" blocks, for a total of 524,800 IP addresses ((8 x 65536) + (2 x 256) = 524,800). Though recent discovery efforts show that only roughly 50-60,000 devices are present in this space at any given time, and even that many of them are statically persistent, a great amount of dynamism does exist. This dynamism results in part from the relatively autonomous apportioning of address blocks in an extremely decentralized organization: there is no central registry that can be consulted as to the use of any given IP address, or even any given address range.

2.1.1 Issues of Scale

Experience shows that the industry's "best of breed" tool for rapid network discovery, nmap [4], will require anywhere from a fraction of a second to several minutes per address in a sequential sweep, depending upon the scanning options employed and the number of ports tested. A simple ICMP "ping" sweep may execute over hundreds of addresses per minute, whereas a limited scan of all IANA-assigned TCP or UDP ports ($<$1024) may take several minutes per host. Even discounting the time required to sequentially locate 50,000 live addresses out of roughly 500,000 possibilities, at the rate of 2 minutes per host, such an inspection would require 1,666 consecutive hours to complete. If, through the judicious tuning of such an effort, the time-per-live-host could be reduced by even an order of magnitude — to twelve seconds on average — the effort would still require nearly a full week's time, running non-stop.

To complicate matters, during the course of a normal workweek, the overwhelming majority of devices are down more often than they are up. By organizational policy, desktop and laptop PCs are turned off when not in use (outside of normal business hours). This results in, at best, an eight-hour window during which anything other than dedicated systems (servers, routers, printers, eg.) can be inspected during any 24-hour period — and of course weekends and holidays preclude this ability as well.

2.1.2 Decentralization and IP Dynamism

Two factors of decentralization within the enterprise further exacerbate the dynamism in the IP pool. The first, as stated above, is that IP address apportioning is not centrally controlled, or even centrally tracked. Each individual business unit is responsible for the assignment of IP blocks and individual addresses. Though a loosely-knit team of enterprise-oriented network engineers collaborate on the WAN infrastructure, and consequently understand the "big picture" with respect to the routed network, no one individual, or even team of individuals, will know what assignments are in place at any given time. As new systems are deployed or decommissioned daily, IP assignments change daily. Further, as new network segments are deployed, or old ones reconfigured, entire blocks may change without notice.

The second exacerbating factor is that the enterprise itself is widely geographically decentralized, with offices ranging from the East Coast to Alaska, and consequently spanning six time zones. The result is that the "eight-hour" workday is greatly reduced from the perspective of a single locale. At 8:00 am EST folks in California are mostly still unplugged, and will be for several hours; likewise by 5:00 pm PST folks on the Eastern

Seaboard have been unplugged for quite some time. Unfortunately, there is no equally decentralized team in place to conduct or coordinate regionally targeted discovery efforts for tactical purposes.

2.1.3 Resultant Known State

The result of the above scenario is that what can be known about the state of the enterprise computer base and infrastructure is limited in accuracy, and limited even more temporally. Tools such as Tivoli Inventory, SolarWinds, and associated enterprise systems management (ESM) approaches can produce a great deal of data, but are difficult if not impossible to leverage for tactical requirements when speed and accuracy are of the greatest importance.

2.2 Current Threat Model

So why is such a timely and accurate tactical ability to identify systems and services so important? Why aren't more strategic, ESM efforts sufficient? Unfortunately the Bad Guys(tm) don't necessarily work at a stodgy ESM pace.

2.2.1 Rapidly Emerging Threats

One recent development that has fundamentally changed the information security landscape is the emergence of threats which propagate at a previously unseen rate. The Melissa virus is probably the seminal instance of such virulent malware, but the world has since seen a steady increase in both the virulence and pathological impact of these threats. Furthermore, judging from the rapidly proliferating strains of the more commonly encountered worms and viruses, the community of attackers continues to grow.

Both CodeRed and Nimda demonstrated appalling improvements in propagation techniques, even though — or perhaps especially because — social engineering remains a core component of such attacks. Yet far more virulent techniques have been postulated [9], and will likely not remain theoretical for long (if they do so remain).

Yet another class of "zero-day exploits" can be seen to be emerging, arguably to some degree as a result of no- or delayed-disclosure policies by some software vendors. Such commercial industry practices that obscure or delay the disclosure of vulnerabilities to consumers decrease the window of time between patch deployment and

exploit deployment, as it must be assumed that in at least some circumstances such vulnerabilities are discovered in parallel by both the vendor and the attacker. This can be evinced by the increasing number of CERT/CC and other such announcements that simultaneously declare that exploit code for the vulnerability described is already known to be circulating ”in the wild”. The most recent Sendmail bug is a classic example of this problem; by the time the vulnerability was widely announced, the probability was deemed high that exploits against common platforms were already underway [2].

2.2.2 Need for Rapid, Tactical Response

Strategic deployment of both gateway- and host-based antivirus technologies is a must; there is no question in this respect. But in even the best-managed enterprise environments, it can fail. Individual systems can exhibit software failures that result in out-of-date signature files, or disabled protection. Clever attachment strategies can defeat gateway countermeasures (as can non-corporate email clients and accounts). Equally, timely patching of vulnerable systems is no longer merely good practice; it is critical. It is also prone to failure.

In the realm of malware, multi-vector propagation is now the norm, which means that once a perimeter has been breached, otherwise protected systems may be at risk. And propagation can now outstrip even the most rapid signature deployment strategies, particularly in very large environments. Epidemics can emerge in advance of all early warning systems in many enterprises, calling for extremely rapid containment measures. The obvious containment measure in such instances is to immediately isolate from the network any and all infected systems, as soon as they can be discovered.

With respect to non-malware threats, a highly tactical response is frequently warranted as well. Internet-facing systems and services are under constant siege, and the response to the next ”zero-day” exploit remains a window of time away — and that window is probably shrinking for reasons discussed above. Such a tactical response must at a minimum leverage the most common techniques used in the attacker community, and network-based reconnaissance has for a decade remained the critical first step [3] [1].

2.2.3 Need for Rapid, Accurate Recon

Unfortunately, due both to the virulence of recent strains of malware, and the emergence of other ”zero-day” exploits, containment is impossible without an almost imme-

diate identification of both infected or compromised hosts, and vulnerable ones. This translates into an immediate ability to identify each and every host listening on a tell-tale backdoor port (where trojans can be identified), or an identical ability to discover each and every host listening on the port under attack (and perhaps to identify the OS and server software on the listening end, down to specific versions).

2.3 Resulting Risk

Based on the scenario above, this enterprise was at an unusually elevated level of risk for incidents where containment, aided by tactical reconnaissance, was critical. Lacking efficient and effective tools that scale to the size and distribution of the largest networked environments, very large-scale enterprises will continue to exhibit this exposure.

3 The Solution

Some aspects of a solution to the above problem should almost certainly, by now, be as obvious to the reader as they were to the author. One approach to assuage the temporal constraints within the environment might be to delegate regional responsibility for such tactical discovery efforts (though coordinating them nationally). Unfortunately, as stated above, the apparatus for such a team effort does not immediately exist, though it could be built over time. Unfortunately, an immediate need exists. Alternately some manner of parallelization might be successfully employed to enable a single, central effort; this was the first improvement attempted.

3.1 Massive Parallelization

3.1.1 nmap Limitations & Solutions

nmap does provide some parameters for adjusting the parallelism employed, but these appear to only affect the number of source ports employed for any given target [4], rather than the actual parallelizing of targets themselves in a multi-threaded way. Further experience shows that use of the maximum recommended parameters (–min-parallelism 100 –max-parallelism 900) does not reduce the per-host threshold substantially for tightly targeted tests (involving only a small number of ports). Consequently, in the given environment, even the most limited of serial discovery efforts have required in excess of

48 hours to run to completion.

Two general approaches to a solution were considered here. The most elegant and computationally efficient might have been to enhance nmap with native multi-threading capabilities. Unfortunately, your author does not speak C++ with any utile fluency, and while a single platform of choice would have sufficed, he is equally unfamiliar with POSIX threads at the level required for such work.

The second, more expedient approach essentially involved perl-scripting a controlled fork() brigade of nmap instances. The basic idea was to programmatically enumerate the 2,048 ”Class C” constituents from the eight ”Class B” blocks, and to scan each of these with a separate nmap instance, to some configurable maximum concurrence. This approach has been remarkably successful: more than two orders of magnitude of concurrence has been reliably achieved from a single point of inspection, resulting in far better than an order of magnitude in the reduction of time required to explore the entire enterprise address space.

3.1.2 Hardware Limitations & Network Impact

The initial assumption was that such an effort would ultimately be bound by network capacity, as the discovery activities saturated all proximal WAN links, or that it would substantially impact network resources at a bare minimum. This did not turn out to be the case; the network traffic generated by such discovery techniques is so small that it remains negligible even increased by several orders of magnitude. It should be noted, however, that the testing platform is in a relatively topologically central location, that the spokes run at a minimum of 1.5Mbps capacity, and that they average no more than 75-85

Somewhat unexpected was the degree to which such an approach would become CPU-bound. Indeed, a 1.8GHz Intel Pentium 4 system (running Linux and installed with 1.5GB of RAM) has been seen to be able to support no more than 200-250 concurrent nmap instances, setting an upper bound on parallelism due to CPU utilization (memory usage was not a factor). However, the design described above would lend itself to the full employment of multiple processors, and consequently the use of an SMP system for future efforts is being arranged.

3.2 Adaptive, Intelligent Searches

3.2.1 False Positives and Differential Port Selection

Typically, the use of such tools and techniques has involved the search for a single port or service on a single platform — for example, port 25/tcp on UNIX for Sendmail, or 1434/udp on Windows for SQL Server. nmap makes such a task fairly trivial in most circumstances, as its operating system fingerprinting capability can be quite effective.

However, TCP fingerprinting is relatively time consuming as it requires multiple connection attempts across multiple ports per host. Moreover it is computationally expensive, as packets must be crafted (requiring a divergence in the logic of the native TCP stack), and their results compared against a large and growing database of signatures. The result is that its use is not always expedient.

Instead, ”differential” port selection can be used. Both UNIX and Windows are capable of serving SMTP, so the presence of a listener on port 25/tcp is obviously not a positive indicator of the likelihood of a Sendmail instance. Consequently, scanning for ports 25/tcp and 445/tcp is a more useful approach, as we can then query the results for instances where 25/tcp is open and 445/tcp is not (port 445/tcp is not commonly used by UNIX, but is open by default in Windows 2000 and above).

Likewise, it was discovered that many of the printers in the described environment appear to be ”open” on all ephemeral udp ports, so merely scanning for 1434/udp will yield a very large percentage of false positives. Adding port 515/tcp to filter against effectively addresses this problem, as Windows systems are unlikely to be used as non-SMB print servers.

3.2.2 Multiphased Approaches

In the case of scanning for 1434/udp to locate all SQL Servers in advance of the onslaught of the SQLSlammer worm, a single discovery effort was sufficient. SQL Server is not an enterprise-supported or -deployed technology in this environment, so locating the small number of ”rogue” instances for patching was straightforward.

Conversely, locating all Cisco routers might require a multiphased approach, because differential port selection is less effective where telnet or ssh are the only reliably present services, and many different brands of routers (or many other hosts) may be configured this way. In this instance it is more effective to conduct an ICMP-based ping sweep in order to locate all live addresses (culling the 50,000 from the 500,000), and

then immediately begin TCP fingerprinting the smaller space of addresses. When executed in parallel over such a smaller space — and against a target population that is likely to be ”always on” — such efforts can be both rapid and accurate.

3.2.3 Advanced Techniques

In many circumstances, even multiphased, concurrent nmap scans are not sufficient to identify target services. For example, merely locating all web servers running on Windows machines did not yield a list of potential CodeRed targets, as Apache W32 instances showed up among the Microsoft IIS instances. Likewise, locating SMTP listeners on UNIX systems may not be helpful unless the specific server and version can be established.

To further aid in the tactical discovery effort in such cases, netcat [5] can be employed in a similar fashion to automate a short exchange with each listening daemon, sufficient to identify the server software — a technique known as ”banner grabbing”. The tools described above can be leveraged to expedite this technique as well, to considerable success.

Finally, the reader might reasonably infer from the above discussion that the environment studied is not internally filtered or firewalled. Were filtering to be deployed, further adaptive techniques would be required.

3.3 Future Developments

As discussed above, the intent is to increase the hardware horsepower behind the tools and techniques discussed, which is expected to yield considerably improved capabilities. Also, the nmap output format currently in use is the deprecated -oM (”machine parseable”) option, and all querying of the data is done by with grep invocations (which requires a certain skill set). The next major improvement will consequently consist of a retooling to parse the newer XML output, and to then automatically store the data in an SQL database, where complex queries are more easily facilitated, and available to a larger community.

4 The Result

Based on the geographical constraints discussed above, as well as the typical work habits of the enterprise's employee base, it has been determined that any discovery efforts targeting systems that are only available during working hours must be confined to a window between 1:00 and 4:00 pm Eastern (8:00 to 11:00 am Alaska) — a mere three hours. Planned improvements notwithstanding, the current tools as deployed provide the capability to conduct such narrowly focused reconnaissance across the enterprise's entire IP space within roughly 2 to 3 hours.

These tools and techniques have been employed many times over the past several years to combat rapidly emerging threats as they have appeared. As a result, the enterprise studied has managed to dodge a number of bullets that struck other similar enterprises to disastrous effect. Some of these are discussed below.

4.1 CodeRed, et al.

First discovered in July of 2001, the CodeRed worm and its subsequent variants ripped through the entire Internet, exploiting a well known vulnerability in the Microsoft IIS webserver. Its highly virulent nature and the widespread availability of vulnerable targets combined to create the single largest information security disaster seen to that day. It was far more devastating in its impact that any previous outbreak, largely because it opened a trivially exploitable backdoor on every infected system, allowing the subsequent execution of arbitrary code, in privileged context, by anyone in the world at any time. Hundreds of thousands of systems were compromised within a single day [7], and entire enterprises were disabled.

IIS is not an enterprise-supported or -deployed technology in the organization studied, but reconnaissance efforts at the time showed that more than a hundred "rogue" instances of the software were running inside the perimeter. Once discovered, the unpatched systems were rapidly taken offline, inspected for compromise, and IIS either patched or disabled before being returned to duty. As a consequence of rapid tactical response to this threat, only three compromises were suffered in the entire enterprise.

4.2 SQLSlammer

The SQLSlammer worm was another extremely prolific strain, this time attacking the Microsoft SQL Server. As stated above, while not an enterprise-supported technology, well over a hundred instances of this server were quickly located and either patched or

disabled. Whereas at least one major bank and several major airlines suffered debilitating outages due to the infestation [8], the enterprise in question sustained not a single infection.

4.3 Remote Buffer Overflow in Sendmail (CA-2003-07)

Unlike the previous two threats discussed, Sendmail is a widely deployed server in the described enterprise environment. At the time CA-2003-07 was released, well over a thousand vulnerable instances were running on a variety of platforms, distributed across the entire enterprise. Using the techniques described above, within hours of the announcement, a comprehensive list was generated including the version and platform of each Sendmail instance. This list was then used to track down and disable all non-corporate and other unnecessary instances, and to secure what remained. Again, no compromises resulting from this vulnerability have been seen in the studied enterprise.

5 Conclusion

In addition to the above three examples of rapidly emerging threats, many more have come about since, and one can only presume that many, many more will be seen hence. A proactive, defense-in-depth posture is indispensable to the security of the enterprise infrastructure and assets, but it is not sufficient. Even the tightest security posture can suddenly be found lacking in the face of unanticipated threats, making rapid tactical capabilities for reactively dealing with such events equally important.

Unfortunately, very large-scale, decentralized enterprises face unusual challenges where reactive capacities are concerned. Happily, these can be overcome at a relatively small cost in terms of time, equipment, and skill.

Finally, all preliminary results indicate that the above techniques can successfully be employed to conduct external perimeter testing (and reconnaissance) at similar scales. As your author is not sufficiently deluded to imagine that he is the first person to consider or attempt such an approach, we should consider the viability of the discussed techniques to be further evidence of the viability of future near-zero-day threats.

References

[1] Boulanger, A., "Catapults and Grappling Hooks: The Tools and Techniques of Information Warfare", *IBM Systems Journal*, 37(1): 106-114 (1998), URL: http://www.research.ibm.com/journal/sj/371/boulanger.html; accessed June 28, 2003.

[2] CERT/CC, "Cert Advisory CA-2003-07 Remote Buffer Overflow in Sendmail", Original release March 2003, Last revised June 2003, URL: http://www.cert.org/advisories/CA-2003-07.html; accessed June 29, 2003.

[3] Cheswick, William R., and Bellovin, Steven M., *Firewalls and Internet Security: Repelling the Wily Hacker*, Addison Wesley Publishing Company, 1994, pp. 145-148.

[4] Fyodor, et al., *nmap* source code and documentation, v3.29, June 2003, URL: http://download.insecure.org/nmap/dist/nmap-3.28.tgz; accessed June 28, 2003.

[5] Giacobbi, Giovanni, et al., *netcat* source code and documentation, v0.6.1, June 2003, URL: http://telia.dl.sourceforge.net/sourceforge/netcat/netcat-0.6.1.tar.gz; accessed June 28, 2003.

[6] Grime, Richard, "Implementing Vulnerability Scanning in a Large Organisation", SANS GIAC, June 2003, URL: http://www.giac.org/practical/gsec/Richard_Grime_GSEC.pdf; accessed June 28, 2003.

[7] Moore, David, "The Spread of the Code-Red Worm (CRv2)", CAIDA, 2001, URL: http://www.caida.org/analysis/security/code-red/coderedv2_analysis.xml; accessed June 29, 2003.

[8] Sieberg, Daniel, and Bash, Dana, "Computer worm grounds flights, blocks ATMs", CNN, January 26, 2003, URL: http://www.cnn.com/2003/TECH/internet/01/25/internet.attack/; accessed June 29, 2003.

[9] Weaver, Nicholas C, "Warhol Worms: The Potential for Very Fast Internet Plagues", February 2002, URL: http://www.cs.berkeley.edu/~nweaver/warhol.html; accessed June 28, 2003.