

IBM Software Group



 e-business software

# Network-based vulnerability assessment

*Pier Luigi Rotondo*

IT Specialist

IBM Tivoli Rome Laboratory

# Abstract

---

Vulnerability assessment aims at identifying weaknesses and vulnerabilities in a system's design, implementation, or operation and management, which could be exploited to violate the system's security policy. The overall scope of vulnerability assessment is to improve information and system security awareness by assessing the risks associated. Vulnerability assessment will set the guidelines to close or mitigate any risk and reinforce security processes. Furthermore it will form an auditable record of the actions performed in protecting from the most current vulnerabilities.

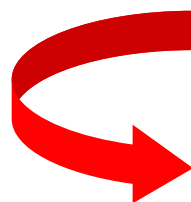
The purpose of a *network-based vulnerability assessment* is to identify the weaknesses and vulnerabilities visible and exploitable on the network.

This presentation describes a complete methodology of network-based vulnerability assessment.

# Vulnerability assessment

- The discipline of vulnerability assessment comprises *host-based vulnerability assessment*, related to the inside configuration of a host, and *network-based vulnerability assessment*, focused on the vulnerabilities visible and exploitable on the network.
- Both kinds of vulnerability assessment are required for maximum effectiveness, as vulnerabilities can be exploited by an entity inside the security perimeter (i.e. a legitimate user), or initiated from outside the perimeter, by an unauthorised or illegitimate user.

**Host-based  
vulnerability  
assessment**



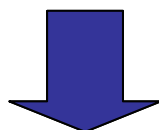
**Network-based  
vulnerability  
assessment**



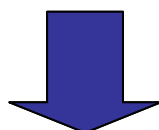
# Anatomy of a network-based vulnerability assessment

- The purpose of a network-based vulnerability assessment is to **compile an inventory of systems and services attached to the network** and, for each system and service, **identify the weaknesses and vulnerabilities visible and exploitable on the network**, also taking advantage of the attacker's techniques. The activity aims at remotely assessing a network by finding vulnerabilities on its systems. Results are eventually consolidated in a **report**.

target acquisition



vulnerability assessment



reporting

target acquisition



vulnerability assessment

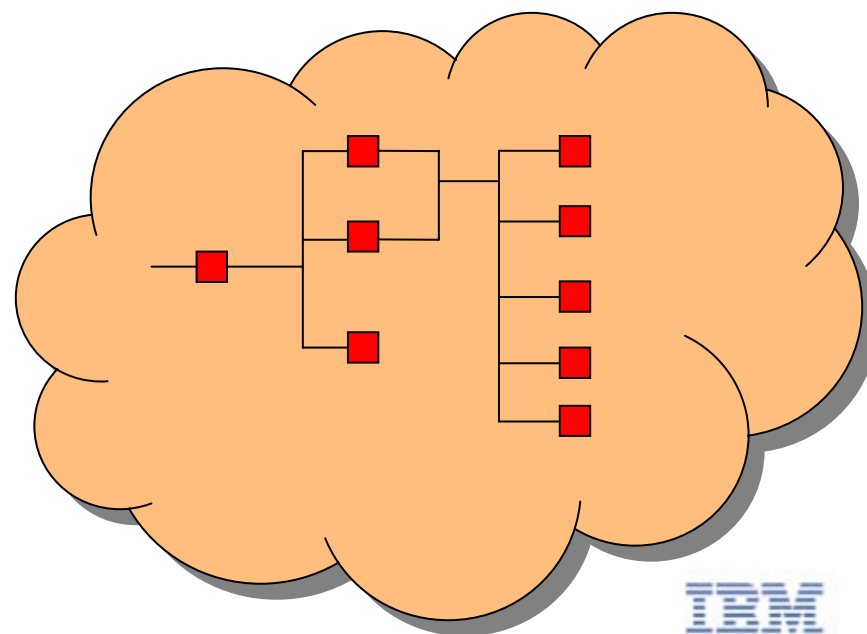


reporting

# Target acquisition

- To acquire complete knowledge of the network environment to analyse, identifying all the alive hosts and network-attached devices (including network equipments, network printers, etc.) residing on the portion of network under analysis, along with their available services.
- Required to outline the baseline for all the subsequent security activities and clearly *define and narrow the scope of your network-based vulnerability assessment*.

security activities



# Network mapping

---

Many different ways (as testing may be performed under various constraints and conditions):

1. Starting from the network topology .... when available ;-)
  2. System-provided tools and information (i.e. ping and traceroute, ICMP queries, routing tables, DNS interrogation with nslookup, DNS zone transfers, etc.)
  3. Specific tools (i.e. nmap, fping, pinger, etc.)
- Theoretically any layer in the ISO-OSI model can provide useful information.
  - In practice, is carried out mainly by using ICMP, TCP or UDP protocols, combinations of the above protocols, or protocols residing on upper layers.
  - Also referred as *IP scanning*, *host discovery*, etc.

# An example of network mapping

## ■ ICMP protocol:

```
# nmap -sP 10.0.1.1-254
Starting nmap (www.insecure.org/nmap/)
Host puma.mydomain.com (10.0.1.1) appears to be up.
[...]
Host neptune.mydomain.com (10.0.1.115) appears to be up.
Host iron.mydomain.com (10.0.1.216) appears to be up.
Nmap run completed -- 254 IP addresses (18 hosts up) scanned in 27 seconds
```

## ■ TCP protocol (i.e. port 80):

```
# nmap -sP -P0 -PS80 10.0.1.1-254
TCP probe port is 80
Starting nmap (www.insecure.org/nmap/)
Host puma.mydomain.com (10.0.1.1) appears to be up.
Host aqua.mydomain.com (10.0.1.3) appears to be up.
[...]
Host iron.mydomain.com (10.0.1.216) appears to be up.
Nmap run completed -- 254 IP addresses (18 hosts up) scanned in 32 seconds
```

*baseline* for all the subsequent security activities ➡

### Target list:

```
10.0.1.1 #puma.mydomain.com
10.0.1.3 #aqua.mydomain.com
10.0.1.4 #silver.mydomain.com
10.0.1.12 #columbia.mydomain.com
```



# Port mapping

---

- *Port mapping* (also known as *port scanning*) is the process of connecting to TCP and UDP ports of the target system to determine what ports are in a LISTENING state, possibly identifying also the running services.
- More specifically:
  1. Identifying the TCP and UDP ports in a LISTENING state;
  2. Identifying the TCP and UDP services running on the target system;
  3. Identifying RPC registered RPC programs running on the target system;
  4. Identifying services *unintentionally* exposed;

# An example of port mapping (using *nmap*)

- Carried-out with programs known as *port mappers* or *port scanners* (i.e. nmap, strobe, netcat)

```
# nmap -sTU 10.0.1.1
```

```
Starting nmap by fyodor@insecure.org (www.insecure.org/nmap/)
```

```
Interesting ports on puma.mydomain.com(10.0.1.3):
```

Port	State	Service
21/tcp	open	ftp
23/tcp	open	telnet
25/tcp	open	smtp
69/udp	open	tftp
80/tcp	open	http
111/tcp	open	sunrpc
111/udp	open	sunrpc
177/udp	open	xmcp
2049/tcp	open	nfs
2049/udp	open	nfs
6000/tcp	open	X11

*expected running service*  
(IANA numbers)

<http://www.iana.org/assignments/port-numbers>

```
Nmap run completed -- 1 IP address (1 host up) scanned in 13 seconds
```

ports in LISTENING state

# Common port mapping techniques

- TCP connect() scanning: complete TCP 3-way handshake connection (SYN, SYN/ACK, ACK), performed via the *connect()* system call (hence the name). The destination host resets (RST) the connection if the port is closed.

```
12:53:25 source.42444 > target.22: S 2375055378:2375055378(0) win 16384 <mss 1460>
12:53:25 target.22 > source.42444: S 2905201405:2905201405(0) ack 2375055379 win 16060 <mss 1460>
12:53:25 source.42444 > target.22: . ack 1 win 16060
12:53:25 source.42444 > target.22: R 1:1(0) ack 1 win 16060
```

```
12:54:47 source.42445 > target.smtp: S 962808263:962808263(0) win 16384 <mss 1460>
12:54:47 target.smtp > source.42445: R 0:0(0) ack 962808264 win 0
```

- SYN scanning (also known as “*half-open*”): source host sends a SYN packet. The destination host replies with a SYN/ACK if the port is listening, RST otherwise. Less likely to be logged.

```
12:56:35 source.46732 > target.22: S 409413429:409413429(0) win 1024
12:56:35 target.22 > source.46732: S 695417011:695417011(0) ack 409413430 win 16384 <mss 1460>
12:56:35 source.46732 > target.22: R 409413430:409413430(0) win 0
```

```
12:57:27 source.54404 > target.smtp: S 543563440:543563440(0) win 2048
12:57:27 target.smtp > source.54404: R 0:0(0) ack 543563441 win 0
```

# Common port mapping techniques

- Fragmentation scanning: layer 4 packets/datagrams are encapsulated into small IP fragments. As TCP/UDP headers are split across different IP fragments, this method is less likely to be detected by (old-style) packet filters.
- UDP scanning: the source host sends UDP datagrams. If the port is closed the destination host sends back an "ICMP port unreachable" message. Eventually, by exclusion, the listing of UDP open ports can be determined.

12:58:45 source.49764 > target.syslog: **udp** 0

12:59:51 source.56315 > target.515: **udp** 0

12:59:51 target > source: **icmp**: cmvc2000test udp **port 515 unreachable**

- IDLE scanning: the scan involves a dumb zombie host, that bounces packets towards the destination host. Based on predictable IPID sequence numbers. Useful for testing IP-based trust relationship.

## UDP port mapping problem: datagrams retransmission

- Neither UDP datagrams nor returning ICMP errors (encapsulated within IP datagrams) are guaranteed to arrive, therefore to avoid false positives UDP scanners must implement a UDP retransmission mechanism (retransmitting datagrams that appear to be lost) when probing UDP ports.

12:58:45 source.49764 > target.syslog: **udp 0**  
Nothing returned! Question: open port or lost UDP/ICMP datagrams?

- This is a very common problem when probing destination hosts that are many hops away from the probing host. *As side effect, ports may be wrongly reported as OPEN, while in reality they are not.*
- Some programs feature dynamic delay and retransmission, others allow user-settable controls for UDP timeouts and retry. Increasing the number of retries will sometimes help the reliability. Others? *Unknown!*

# Stop unneeded services!

- Remember that none of the services are really secure. Each has its own security weaknesses, and it will be probably exploited by hackers if it hasn't already been exploited.
- Based on the output from the port mapping, identify each listening port along with its associated service and then *stop any service and close any port you don't need, that you do not know, or that your are not sure about.*

*Unneeded active services expose the system to unnecessary risks!*



## Putting things together ...

- Based on the results from *network mapping* and *port mapping* you will end up with a clear picture of the systems and devices on you network together with their active ports.

Address Range 10.0.1.1-10.0.1.254

Detected host puma.mydomain.com (10.0.1.1)

Open TCP ports: 21, 23, 25, 80, 111, 443, 2049, 6000

Open UDP ports: 69, 111, 177, 2049

Registered RPC programs: portmapper (port 111), nfs (port 2049),  
mountd (ports 32887, 33257)

Detected host aqua.mydomain.com (10.0.1.7)

Open TCP ports: 21, 23, 111, 6000, 32768

Open UDP ports: 111, 177

Registered RPC programs: portmapper (port 111), unknown (port  
32997)

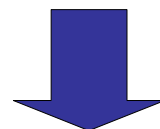
[...]

Detected host iron.mydomain.com (10.0.1.216)

Open TDP ports: 135, 139, 1025, 1026

Open UDP ports: 123, 135, 137, 138

target acquisition



vulnerability assessment



reporting

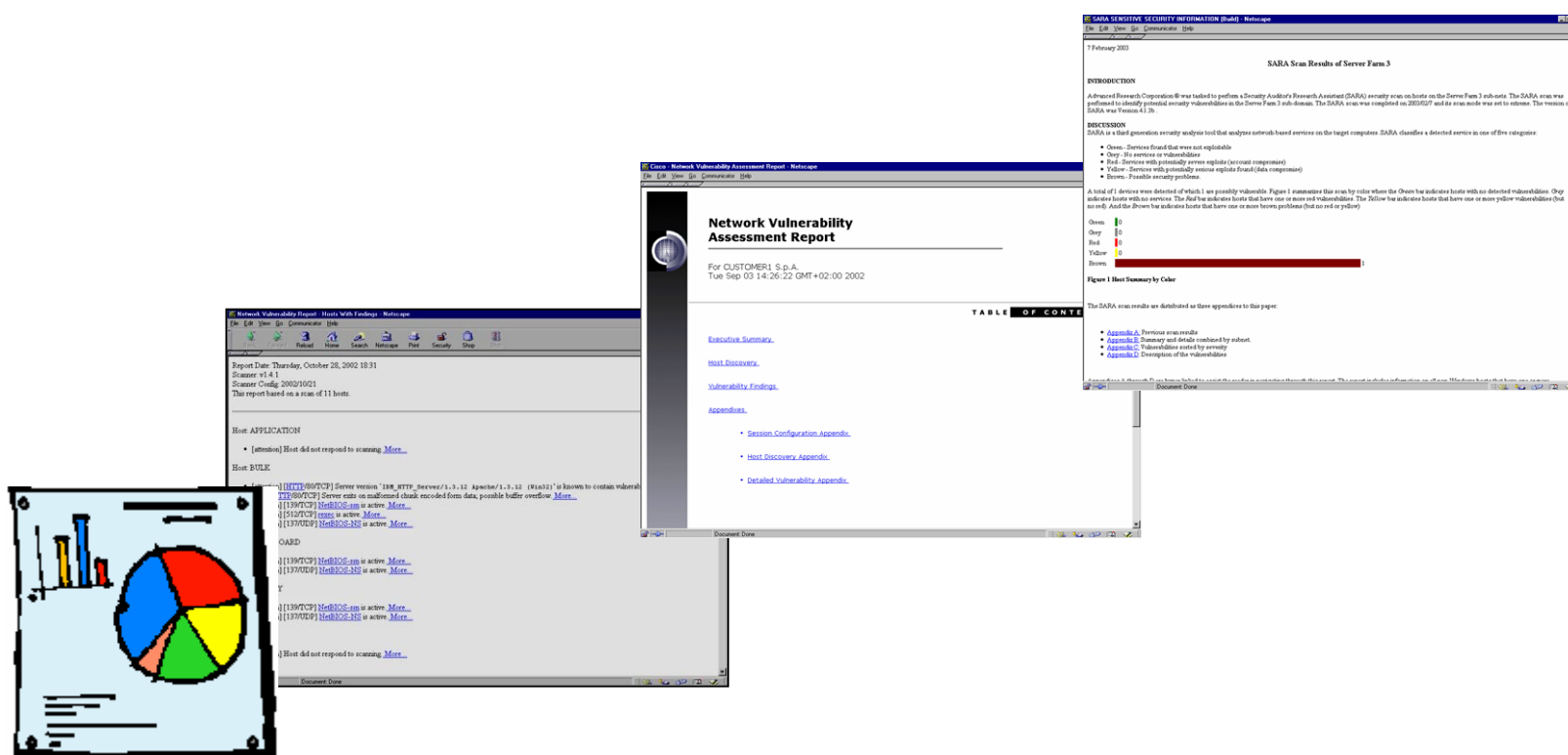


# Network-based vulnerability scanning programs

- Vulnerability scanners take the concept of a port scanner to the next level. The vulnerability scanner identifies not just hosts and open ports but any associated vulnerabilities automatically instead of relying on human interpretation of the results
- Automated network-based vulnerability scanning programs assist in:
  1. extracting information from the target hosts (O.S. version, open ports, active services and protocols, version of each running service, exported resources and shares, valid accounts), this phase is also referred as *enumeration*
  2. checking all the details against publicly available sources of known vulnerability information and vendor security alerts to see if known potential vulnerabilities may affect the host
  3. performing tests and use heuristics to confirm the existence of a real vulnerability (whenever possible and according to the level of aggressiveness chosen)
  4. rating the risk of the vulnerability
  5. mapping each finding to their related security advisory or alert
  6. providing fixing direction
  7. creating reports

# Network-based vulnerability scanning programs

- The output is a deliverable with directions to close, or at least mitigate, the risks associated to each weakness or vulnerability found.



## Available programs

---

- ARC SARA - Security Auditor's Research Assistant
- eEye Digital Security Retina
- BindView
- CyberCop Scanner
- ISS Internet Security Scanner
- ISS Internet Scanner
- Kane Security Analyst
- NA CyberCop Scanner
- Nessus
- Symantec NetRecon
- Saint corporation Saint (formerly WWDSI)
- Satan - Security Administrator Tool for Analyzing Networks
- Vigilante SecureScan NX
- WebTrends Security Analyzer

# ISS

- ISS (Internet Security Scanner) was first released as a shareware product in 1992 by Christopher Klaus. ISS is a program that interrogates all computers within a specified IP address range, determining the security posture of each with respect to several common system vulnerabilities.
- CERT® Advisory CA-1993-14 Internet Security Scanner (ISS)  
  
" [...]The software package, known as ISS or Internet Security Scanner, will interrogate all computers within a specified IP address range, determining the security posture of each with respect to several common system vulnerabilities. The software was designed as a security tool for system and network administrators. ISS does not attempt to gain access to a system being tested. However, given its wide distribution and ability to scan remote networks, the CERT/CC believes that it is likely ISS will also be used to locate vulnerable hosts for malicious reasons. [...]"
- Christopher Klaus formed Internet Security Systems in 1994 and released a commercial product, *ISS Internet Scanner*.

# Satan

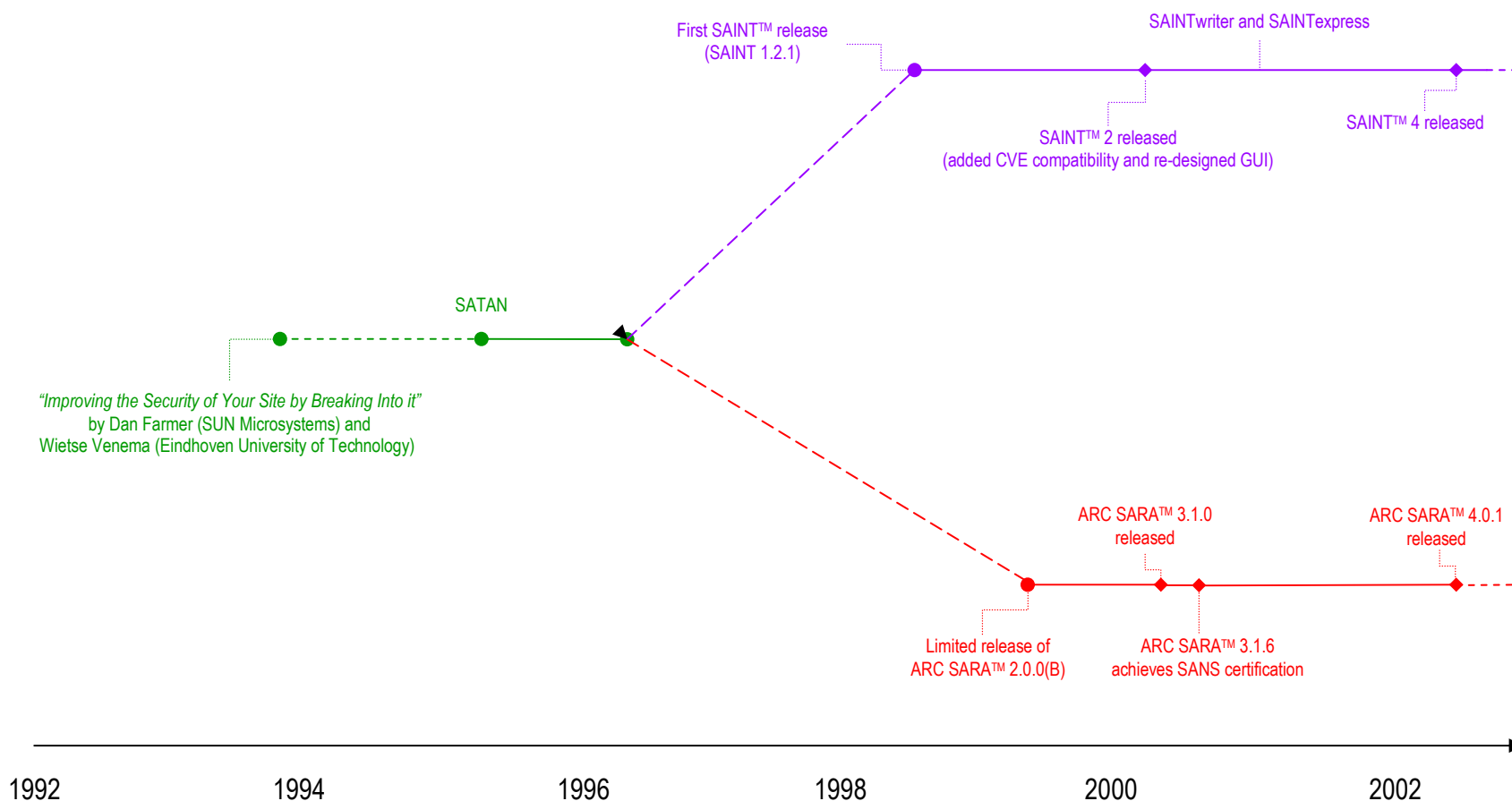
---

- SATAN Security Administrator Tool for Analyzing Networks, written by Wietse Venema & Dan Farmer, and first released on April 1995.
- CERT® Advisory CA-1995-06 Security Administrator Tool for Analyzing Networks (SATAN):

"[...] SATAN was designed as a security tool for system and network administrators. However, given its wide distribution, ease of use, and ability to scan remote networks, SATAN is also likely to be used to locate vulnerable hosts for malicious reasons. It is also possible that sites running SATAN for a legitimate purpose will accidentally scan your system via SATAN's exploratory mode.

Although the vulnerabilities SATAN identifies are not new, the ability to locate them with a widely available, easy-to-use tool increases the level of threat to sites that have not taken steps to address those vulnerabilities. In addition, SATAN is easily extensible. After it is released, modified versions might scan for other vulnerabilities as well and might include code to compromise systems. [...]"

# Evolution of SATAN-based vulnerability scanning pgms



## How to evaluate a good program?

---

- As new exploits and attack schemes come into existence, one important criterion in evaluating a vulnerability scanning program is *the frequency at which the vulnerability database and testing engine are updated*. An old program, or just an old version of a vulnerability scanner, will only test for old and probably already fixed vulnerabilities, while it will not test your systems for new and recent attacks.

# Vulnerability confirmation

---

- Not all potential vulnerabilities will be confirmed as real vulnerabilities during the tests. Some network-based vulnerability assessment programs are able to distinguish between a *potential vulnerability* and a *confirmed vulnerability*.
- Exploit code, often available from public security resources, can be used to confirm the existence of real vulnerabilities.
- Unstructured and manual activity, typically performed by matching information from multiple resources.
- Regardless of the success or failure to exploit a potential vulnerability, the underlying vulnerability may still exist. *Potential vulnerabilities should therefore be treated with the same seriousness as confirmed vulnerabilities.*



## Surface vulnerability and risk rating

---

- A surface vulnerability is a weakness, as it exists in isolation, that is without any other vulnerability.
- The difficulty in rating the risk level of vulnerabilities is that they rarely exist in isolation. For example there could be several *low risk* vulnerabilities that coexist on a particular network that, when combined, present a high risk.
- A vulnerability scanner would generally not recognize the danger of the combined vulnerabilities and thus would assign a low risk to them leaving the network administrator with a false sense of confidence in his or her security measures. The reliable way to identify the risk of vulnerabilities in aggregate is through penetration testing.

# Hierarchy of threats

---

- The question is a little more difficult to answer and takes experience along with knowing how the threat was carried out. Which factors contribute more to a website defacement? The Operating System or the type of web server running on the Operating System?
- We need a hierarchy of threats. Although the threat is still the website defacement, an attacker could use multiple methods to deface a site. The threat probability is then a combination of which methods of attack are the most popular, along with which system configurations are most susceptible to those popular attacks.

# Performance

---

- Network vulnerability scanning is an intensive and time-consuming operation. The amount of time it takes for any assessment program to scan a single host can vary according to several factors, including network performance and the number and type of tests performed on the given host.
- *Distributed scanners* achieve faster results by using a coordinated set of agents that scan well determined partition of the target list each. Results are aggregated into a central repository and therefore a single report can be created on a central console.
- The biggest obstacle is a feature known as *ICMP rate limiting* (defined in RFC1812 “*Requirements for IP Version 4 Routers*”) implemented by many Operating Systems. “*Aggressive*” UDP port mappings may yield unreliable results.



# Security report

---

- Eventually all the findings are to be consolidated in a final *security report*. Different levels of information can be included, depending on the audience: technical details, including directions and fix information, for system administrators; summaries for security managers; and high-level graph and trend reports for executives. The aim is twofold: reports are both input for operative directions and auditing records.
- The value of a vulnerability assessment activity is tied to its ability to assist in the remediation of the vulnerabilities found, therefore the final report shouldn't be just a collection of problems but it must include specific advice on how to close the vulnerabilities.

## Security report (cont.)

- The report should describe at least
  1. scope and methodology
  2. detailed findings and directions for improvements, possibly indexed by risk priority, for the technical personnel. Links to vendor advisories
  3. recommendation to avoid the same findings in the future
  4. high level management reports, possibly including historic trends, giving an overall perspective of an organisation's security posture
  5. general recommendations and conclusions
- A vulnerability assessment does not last forever, rather the final report is strictly linked to the timeframe when the scan was performed. A vulnerability assessment is therefore *inherently not exhaustive and the faithfulness of the final report decreases with time.*
- As people operate computers and networks, and people make mistakes, *the vulnerability assessment must be a periodic and iterative process.*

## Sensitivity of the security report

- If weaknesses or vulnerabilities were found, the report in the wrong hands could be extremely dangerous. A competitor may use it for corporate espionage or for discrediting the rival, a hacker may use it to break into the client's systems or share the report with other hackers increasing the threat, an unfaithful employee may steal sensitive or confidential information.

Host: aqua.mydomain.com

[telnet] User account "root" has weak password "root". Type of vulnerability: weak password

login: root

root's Password: \*\*\*\*

Welcome to aqua.mydomain.com

1 unsuccessful login attempt since last login.

Last unsuccessful login: Fri Feb 28 13:48:41 NFT 2003 on /dev/pts/0 from scanner.mydomain.com

\$

- The entire scan team has the responsibility to ensure the safety of the result information, during and well after the security activity.

# Good candidates for ethical hacking

---

**hacker** *noun* 1. A computer **hacker** is someone who tries to break into computer systems, especially in order to get secret or confidential information that is stored there.

(Collins COBUILD English dictionary)

- “*Ethical*” hackers employ the same tools and techniques as “*criminal*” hackers, but they would neither damage the target systems nor steal information. Instead they evaluate the target systems’ security and report back to the owners with the vulnerability they found and instructions on how to remedy them.
- *Ethical hackers must be completely trustworthy.* While conducting security tests, the ethical hacker may discover information that should remain secret, and therefore must be trusted to exercise tight control over any information about a target that could be misused.