

Vulnerability scanning

Security and Privacy
VSE Prag, 7 - 11.6.2010

Institute for Information Processing and
Microprocessor Technology (FIM)
Johannes Kepler University Linz, Austria

E-Mail: sonntag@fim.uni-linz.ac.at
<http://www.fim.uni-linz.ac.at/staff/sonntag.htm>



- Source files
 - shadow1, shadow2, shadow3
 - » Linux password files
 - passwords.txt
 - » A file with passwords extracted from Windows
- Requirements:
 - Administrative rights
 - » For installing software
 - Installed software (see CD)
- Software:
 - Nmap
 - (Nessus) – Must be downloaded and activated!
 - John the Ripper
 - Ophcrack



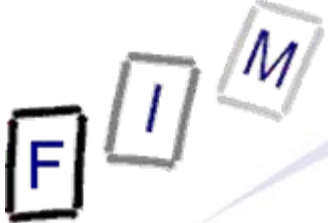
Please note!

- We are not going to attack anyone here!
- We are trying to indentify problems for later fixing it
- Permission is **always** required for vulnerability scanning
 - Which system(s)
 - At what time
 - What kinds of scans (destructive, ...)

We will scan **our own** system here **ONLY!**



- NMap (Network MAPper) is a network scanner
 - It tries to find all computers in a specific network and checks, what ports are open, what OS they are running, whether there is a firewall, etc.
- It does not look for specific vulnerabilities!
 - But it gives recommendations; e.g. services to disable
 - Some scans + vuln. systems → Lock-up/crash!
- Used as a tool for inventory generation in a network
 - Are there any computers which should not be there?
 - Can also be used to gather information for a later attack
 - » Which OS/software and which version is running
- Stages: 1 =Host discovery, 2 =Port scan, 3 =Service/version detection, 4 =OS detection, 5 =Scripting
 - Scripting may also include vulnerability/malware detection!



- Usage:
 - Start program and enter IP address
 - Select profile for scanning
 - » Special options only available in the command line version or when constructing a new profile!
- Your tasks:
 - Install NMap
 - Scan the local subnet for hosts
 - » Use a "Regular scan"
 - Scan the machine of your neighbour
 - » Use a "Operating System detection scan"
 - Interpret the results
 - » Correct output?
 - » Something surprising/dangerous found?

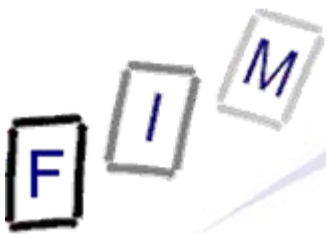


- Password cracking tool
 - Uses word lists as well as brute-force
 - » Word lists can be "multiplied" by mangling rules (reverse, ...)
 - Note: Long lists take longer, but provide better chances!
 - » Brute force: Define character set and set password length limit
 - Can also be used as password-strength checking module
 - "Reconstructs" the password from its hash
 - » Therefore requires access to the password file!
 - Can be interrupted and restarted (may take a long time!)
- Supported are the following password hash types
 - crypt(3) hash types: traditional & double-length DES-based, BSDI extended DES-based, FreeBSD MD5-based (also used on Linux, Cisco IOS), OpenBSD Blowfish-based (also used on some Linux distr.), Kerberos/AFS, Windows NT/2000/XP LM DES-based
 - » More with additional patches!



John the Ripper

- Your tasks:
 - Run John the Ripper against the provided shadow files
 - » "Scenarios/shadow1": Try wordlist
 - » "Scenarios/shadow2":
 - Try wordlist
 - Try incremental (=brute force) search, profile "alpha"
 - » "Scenarios/shadow3": Try in your spare time!
 - Interpret the results/success probabilities



- Password cracking tool for Windows
 - LAN Manager/NT LAN Manager hashes (i.e. Win passwords)
 - » LM / NTLM hashes (not stored in cleartext, but as hash only)
 - » Windows Vista has the (easier) LM hashes disabled by default
 - Older versions still store the weak LM for backwards compatibility
 - Can import the hashes from various formats or read it directly
- Based on Rainbow tables and brute force
 - Some are freely available, others cost money
 - » You could theoretically create them yourself, but this is an extremely time and resource-intensive action!
 - Free tables: About 99.9 % coverage for alphanumeric passwords of up to 14 characters (LM), 99% for NTLM
 - » All printable chars/symbols/space (NT/Vista); German →á US\$ 99



Rainbow tables

- Reducing time by investing memory
 - "Pre-computed passwords"
- Simplest form: Generate all passwords + their hashes and store them for later lookup (immediate cracking!)
 - Drawback: Gigantic table!
- Rainbow tables: Compute all passwords, but store only a small part of them → After finding the hash, some time is required to obtain the actual password
 - Time is reduced by the square of the available memory
- Countermeasure: Use "salting"
 - A random value is generated, prepended to the password, and stored
 - Rainbow table would have to be enlarged for the salt
 - » 4 char salt + 14 char password → 18 char rainbow table!



- Your tasks:

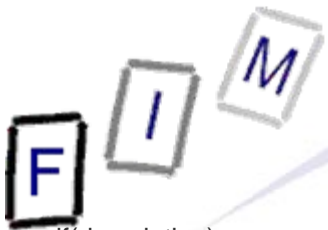
- Run Ophcrack against the provided passwords
- Discuss the results:
 - » Why are some found quickly, but the same password takes much longer in another instance?
 - » Why is this working in Windows, but not for other systems?



- Nessus is a scanner for vulnerabilities
 - Based on signatures → Finds only known problems!
 - » Currently about 35000 plugins
- Updating the signatures: Possible
- First step: Identify OS → Almost all vuln. depend on this
 - Registry, SNMP, ICMP, MSRPC, NTP
- Second step: Check which vuln. might apply and test them
 - Not by actually exploiting them, only whether it would work!
- From where to run the scan?
 - Outside: Probably already safe, best to be sure
 - Inside (Critical machines): Defence in depth
 - DMZ: One computer was hacked → Others still secure?
- Commercial use/additional functionality → You have to pay!
 - US\$ 1200 per scanner per year



- Nessus is separated into a daemon and a client
 - Scanning is done by the daemon(s); the client is just an UI
 - Can do more intensive scanning if provided credentials for logging on to a computer
- Vulnerabilities are scripted in NASL
 - Nessus Attack Scripting Language (see next page)
 - » You can write your own too!
 - Detection is not perfect: False positives may occur
- Attention: Some scans can crash the target!
 - Take care before enabling "all" scans!
 - Option "Safe checks" disables anything dangerous and checks through banners only; no actual trying
- Found a vulnerability? Fix it!
 - Prioritize the problems detected
 - Bugtraq ID or CVE number for obtaining further information



NASL example (phpcms_xss.nasl)

```
if(description)
{
  script_id(15850);
  script_version("$Revision: 1.5 $");
  script_cve_id("CVE-2004-1202");
  script_bugtraq_id(11765);

  script_name(english:"phpCMS XSS");

  desc["english"] = "
The remote host runs phpCMS, a content management system
written in PHP.

This version is vulnerable to cross-site scripting due to a lack of
sanitization of user-supplied data in parser.php script.
Successful exploitation of this issue may allow an attacker to execute
malicious script code on a vulnerable server.

Solution: Upgrade to version 1.2.1pl1 or newer
Risk factor : Medium";

  script_description(english:desc["english"]);
  script_summary(english:"Checks phpCMS XSS");
  script_category(ACT_GATHER_INFO);
  script_copyright(english:"This script is Copyright (C) 2004 David Maciejak");
  script_family(english:"CGI abuses : XSS");
  script_require_ports("Services/www", 80);
  script_dependencie("http_version.nasl", "cross_site_scripting.nasl");
  exit(0);
}
```

```
include("http_func.inc");
include("http_keepalive.inc");

port = get_http_port(default:80);
if ( ! get_port_state(port))exit(0);
if ( ! can_host_php(port:port) ) exit(0);

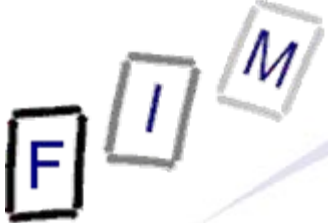
if ( get_kb_item("www/" + port + "/generic_xss") ) exit(0);

buf = http_get(item:"/parser/parser.php?file=<script>foo</script>",
port:port);
r = http_keepalive_send_recv(port:port, data:buf, bodyonly:1);
if ( r == NULL )exit(0);

if(egrep(pattern:"<script>foo</script>", string:r))
{
  security_warning(port);
  exit(0);
}
```



- Usage:
 - Start program and add IP address
 - Select scan policy for scanning
 - » Use the "Default scan policy"
 - You can construct your own as well, e.g. only specific tests
 - Hint: Disable Windows Firewall on both source and target!
- Your tasks:
 - Install Nessus
 - » Register for a home-user activation code
 - » Download the plugins
 - » Reboot or manually start the Nessus service ("Tenable Nessus")
 - Scan the machine of the instructor
 - Interpret the results
 - » Any weaknesses found?
 - » Are they serious?



Conclusions

- Many tools for administrators to improve security can also be used by attackers
 - And they will be...
 - Note: Most security tools leave clear traces in log files
 - » But many can also be used/configured to be quite stealthy!
- Therefore they should be used to test your own systems
 - Regularly, if possible, and with updated signatures
- Set minimum requirements for passwords
 - Or they can be cracked rather easily!
 - » One reason for the shadow passwords file → /etc/passwd is (and needs to be) world readable
 - Especially important on Windows
 - » Also disable the weak LM hashes

Do it yourself, before they do it to you!

F I M

Questions?

Thank you for your attention!