

Project 1

- ∅ Adhere to submission guidelines
 - ∅ Submit all and only the required files
 - ∅ .c, .h, Makefile for each folder – supercipher, togglecipher and losscipher
 - ∅ Don't submit project files from Visual Studio or exe files
 - ∅ Submit PS or PDF of your writeup
 - ∅ Submit Makefile
 - ∅ Make sure your project compiles on EECIS Unix machines
- ∅ For losscipher address both bit loss and block loss

1

Makefile

- ∅ Tutorial
 - ∅ http://www.gnu.org/manual/make/html_chapter/make_toc.html
- ∅ File that holds rules for your program compilation
- ∅ Example:
 - ∅ You have two .c files and three .h files: cipher.c functions.c functions.h cipher.h allincludes.h
 - all: cipher.c functions.c functions.h cipher.h allincludes.h
 - gcc -o supercipher -Ifunctions.h -lcipher.h -Iallincludes.h \
 - tab → -lm functions.c cipher.c → Output file name
 - ∅ If you use math.h


2

It Is Time To Open Emulab Account

- ∅ Go to <http://www.emulab.net>
- ∅ Sign up with *existing* project CIS662 (unfortunately I asked for a wrong name)
- ∅ Go to *Documentation* link and read about how to use Emulab

3

Disclaimer

- ∅ Some techniques and tools mentioned in this class could be:
 - ∅ Illegal to use 
 - ∅ Dangerous for others – they can crash machines and clog the network
 - ∅ Dangerous for you – downloading the attack code you provide attacker with info about your machine
- ∅ Don't use any such tools in real networks – especially not on EECIS network
 - ∅ You can only use them in a controlled environment

4

Intrusions

- ∅ Why do people break into computers?
- ∅ What type of people usually breaks into computers?
- ∅ I thought that this was a security course. Why are we learning about attacks?

5

Intrusion Scenario

1. Reconnaissance
2. Scanning
3. Gaining access at OS, application or network level
4. Maintaining access
5. Covering tracks

6

Phase 1: Reconnaissance

- Ø Get a lot of information about intended target:
 - Ø Learn how its network is organized
 - Ø Learn any specifics about OS and applications running

Low Tech Reconnaissance

- Ø Social engineering
 - Ø Instruct the employees not to divulge sensitive information on the phone
- Ø Physical break-in
 - Ø Insist on using badges for access, everyone must have a badge, lock sensitive equipment
 - Ø How about wireless access?
- Ø Dumpster diving
 - Ø Shred important documents

Web Reconnaissance

- Ø Search organization’s web site
 - Ø Make sure not to post anything sensitive
- Ø Search information on Usenet postings
 - Ø Instruct your employees what info should not be posted
 - Ø Find out what is posted about you
- Ø Use Google to find all documents mentioning this company to find out partner companies
 - Ø Find out what is posted about you

Whois databases

- Ø When an organization acquires domain name it provides information to a registrar
- Ø Looking at public registrar files one can find out:
 - Ø Registered domain names
 - Ø Domain name servers
 - Ø Contact people names, phone numbers, E-mail addresses

Whois databases

<http://www.networksolutions.com>

Domain Name: UDEL.EDU

Technical Contact:
Same as above

Registrant:

University of Delaware
192 South Chapel Street
Newark, DE 19716
UNITED STATES

Name Servers:
DNS1.UDEL.EDU 128.175.13.16
DNS2.UDEL.EDU 128.175.13.17
NOC2.DCCS.UPENN.EDU 128.91.254.1
NOC3.DCCS.UPENN.EDU 128.91.254.4

Contacts:

Administrative Contact:
Daniel J. Grim
Executive Director
University of Delaware
192 South Chapel Street
Newark, DE 19716
UNITED STATES
(302) 831-3700 (302) 831-1990
udel-domain@udel.edu

Domain record activated: 24-Jul-1985
Domain record last updated: 22-Dec-2001

ARIN databases

- Ø Find out range of IP addresses assigned to a company
 - Ø This will be useful later for scanning
 - Ø <http://www.arin.net/whois/arinwhois.html>

ARIN databases

```

OrgName: University of Delaware
OrgID: UNIVER-19
Address: 192 South Chapel Street
City: Newark
StateProv: DE
PostalCode: 19716
Country: US
NetRange: 199.75.219.0 - 199.75.219.255
CIDR: 199.75.219.0/24
NetName: UDEL-219
NetHandle: NET-199-75-219-0-1
Parent: NET-199-75-0-0-1
NetType: Reassigned
Comment:
RegDate: 1996-10-18
Updated: 1996-10-18
TechHandle: RWR3-ARIN
TechName: Reisor, Ron W.
TechPhone: +1-302-831-6030
TechEmail:
    
```

13

Domain Name System

- Ø What does DNS do?
- Ø How does DNS work?
- Ø Types of information an attacker can gather:
 - Ø Range of addresses used
 - Ø Address of a mail server
 - Ø Address of a web server
 - Ø OS information
 - Ø Comments

14

Interrogating DNS – Zone Transfer

```

$ nslookup
Default server:evil.attacker.com
Address: 10.11.12.13
server 1.2.3.4
Default server:dns.victimsite.com
Address: 1.2.3.4
set type=any
ls -d victimsite.com
system1 1DINA 1.2.2.1
         1DINHINFO "Solaris 2.6 Mailserver"
         1DINMX 10 mail1
web      1DINA 1.2.11.27
         1DINHINFO "NT4www"
    
```



15

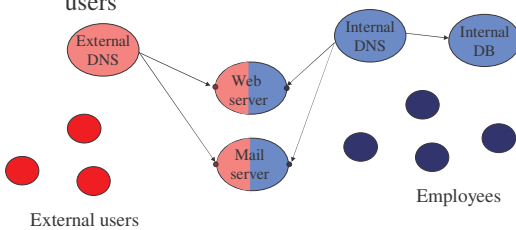
Protecting DNS

- Ø Provide only necessary information – no OS info and no comments
- Ø Restrict zone transfers – allow only a few necessary hosts
- Ø Use split-DNS

16

Split-DNS

- Ø Show different view to external and internal users



17

Reconnaissance Tools

- Ø Tools that integrate ping, whois, ARIN, DNS interrogation and many more services:
 - Ø Applications
 - Ø <http://www.samspade.org/ssw>
 - Ø Web based portals
 - Ø <http://nettool.false.net>
 - Ø <http://www.samspade.org>
 - Ø <http://members.tripod.com/mixersecurity/evil.html>
 - Ø <http://www.network-tools.com>



18

Phase 2: Scanning

- Ø Detecting information useful for break-in
 - Ø Live machines
 - Ø Network topology
 - Ø Firewall configuration
 - Ø Applications and OS types
 - Ø Vulnerabilities

19

War Dialing

- Ø Finding modem access
- Ø Why modems?
 - Ø Networks are protected by a firewall, modems punch holes in firewalls
 - Ø Modem access may not even be password-protected

20

War Dialing

- Ø Find out several phone numbers to feed into a war dialer
- Ø It will try ranges surrounding them
 - Ø Randomly
 - Ø With random pause intervals
- Ø It will record every success, move on if it encounters busy tone or a human picks up
- Ø It takes about an hour to check 100 numbers

21

War Dialing Tools

- Ø THC-Scan
 - Ø Windows Application
 - Ø <http://thc.inferno.tusculum.edu>
 - Ø Easy to use interface
 - Ø Automatic but accepts user input
- Ø TBA
 - Ø PDA application
 - Ø <http://www.l0pht.com>



22

After War Dialing

- Ø Gain access by guessing passwords
- Ø Gain information about OS
- Ø If modem sends a string of characters identifying server application, use specific application client to access it

23

Defenses Against War Dialing

- Ø Do not allow users to install modems
 - Ø Dial-out modems only
- Ø Find your modems before the attackers do

24

Network Mapping

- Ø Finding live hosts
 - Ø Ping sweep
 - Ø TCP SYN
- Ø Map network topology
 - Ø Traceroute
 - Ø Sends out ICMP or UDP packets with increasing TTL
 - Ø Gets back ICMP_Time_Exceeded message from intermediate routers

25

Traceroute Example

```

traceroute to copland.udel.edu (128.175.13.92), 30 hops max, 38 byte packets
 1  dward (131.179.192.1)  0.278 ms  0.288 ms  0.288 ms
 2  131.179.187.3 (131.179.187.3)  0.412 ms  0.431 ms  0.413 ms
 3  Border.CS.UCLA.EDU (131.179.12.1)  0.794 ms  0.808 ms  0.795 ms
 4  compsci-mathsci.backbone.ucla.net (169.232.49.65)  0.642 ms  0.578 ms  0.566 ms
 5  mathsci-core.backbone.ucla.net (169.232.6.109)  0.815 ms  0.659 ms  0.640 ms
 6  core-border.backbone.ucla.net (169.232.6.138)  0.719 ms  0.734 ms  0.867 ms
 7  tus-dcl-ucla-egm.cenic.net (137.164.24.133)  1.693 ms  2.036 ms  1.571 ms
 8  dc-lax-dc2-tus-dcl-pos.cenic.net (137.164.22.42)  2.010 ms  1.803 ms  2.138 ms
 9  hpr-lax-hpr-dc-lax-dc2-gs-2.cenic.net (137.164.22.21)  1.874 ms  2.727 ms  2.383 ms
10  abilene-LA-hpr-lax-gsr1-10ge.cenic.net (137.164.25.3)  16.122 ms  1.835 ms  10.820 ms
11  hstng-losang.abilene.ucaid.edu (198.32.8.22)  33.603 ms  33.774 ms  34.025 ms
12  atlang-hstng.abilene.ucaid.edu (198.32.8.34)  46.541 ms  46.815 ms  46.587 ms
13  washing-atla.abilene.ucaid.edu (198.32.8.66)  73.733 ms  73.866 ms  73.957 ms
14  chp-br4-p-0-0-0.nss.udel.edu (128.175.137.9)  77.276 ms  77.861 ms  77.319 ms
15  chp-rt2-v-9.nss.udel.edu (128.175.111.198)  78.076 ms  77.472 ms  77.515 ms
16  copland.udel.edu (128.175.13.92)  77.266 ms  77.502 ms  77.281 ms
    
```

26

Network Mapping Tools

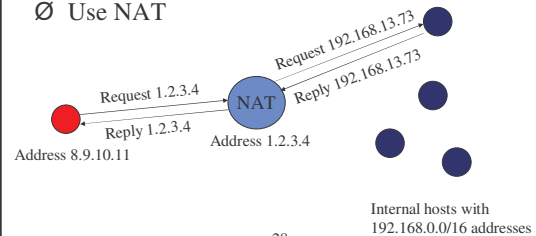
- Ø Cheops
 - Ø Linux application
 - Ø <http://www.marko.net/cheops>
 - Ø Automatically performs ping sweep and network mapping and displays results in GUI



27

Defenses Against Network Mapping

- Ø Filter out outgoing ICMP traffic
 - Ø Maybe allow for your ISP only
- Ø Use NAT



28

Port Scanning

- Ø Finding applications that listen on ports
- Ø Send various packets:
 - Ø Establish and tear down TCP connection
 - Ø Half-open and tear down TCP connection
 - Ø Send invalid TCP packets: FIN, Null, Xmas scan
 - Ø Send TCP ACK packets – find firewall holes
 - Ø Obscure the source – FTP bounce scans
 - Ø UDP scans
 - Ø Find RPC applications



29

Port Scanning

- Ø Set source port and address
 - Ø To allow packets to pass through the firewall
 - Ø To hide your source address
- Ø Use TCP fingerprinting to find out OS type
 - Ø TCP standard does not specify how to handle invalid packets
 - Ø Implementations wildly differ

30

Port Scanning Tools

- Ø Nmap
 - Ø Unix and Windows NT application and GUI
 - Ø <http://www.insecure.org/Nmap>
 - Ø Various scan types
 - Ø Adjustable timing



31

Defenses Against Port Scanning

- Ø Close all unused ports
- Ø Remove all unnecessary services
- Ø Filter out all unnecessary traffic
- Ø Find openings before the attackers do
- Ø Use smart filtering

32

Firewalk: Determining Firewall Rules

- Ø Find out firewall rules for new connections
- Ø We don't care about target machine, just about packet types that can get through the firewall
- Ø Find out distance to firewall using traceroute
- Ø Ping arbitrary destination setting TTL=distance+1
- Ø If you receive ICMP_Time_Exceeded packet went through

33

Defenses Against Firewalking

- Ø Filter out outgoing ICMP traffic
- Ø Use firewall proxies

34

Vulnerability Scanning

- Ø The attacker knows OS and applications installed on live hosts
- Ø He can now find for each combination
 - Ø Vulnerability exploits
 - Ø Common configuration errors
 - Ø Default configuration
- Ø Vulnerability scanning tool uses a database of known vulnerabilities to formulate packets and send them to hosts
- Ø Vulnerability scanning is also used for sysadmin

35

Vulnerability Scanning Tools

- Ø SARA
 - Ø <http://www-arc.com/sara>
- Ø SAINT
 - Ø <http://www.wdsi.com/saint>
- Ø VLAD
 - Ø <http://razor.bindview.com/tools>
- Ø Nessus
 - Ø <http://www.nessus.org>



36

Defenses Against Vulnerability Scanning

- Ø Close your ports and keep systems patched
- Ø Find your vulnerabilities before the attackers do

37

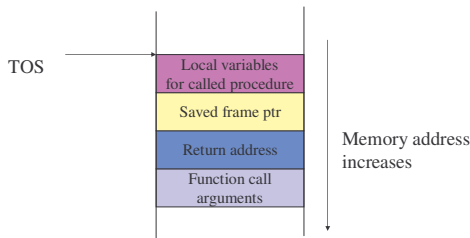
Phase 3: Gaining Access

- Ø Exploit vulnerabilities
 - Ø Exploits for a specific vulnerability can be downloaded from hacker sites
 - Ø Skilled hackers write new exploits

38

Stack-based Overflow Attacks

- Ø Stack stores important data on procedure call



39

Stack-based Overflow Attacks

- Ø Consider a function

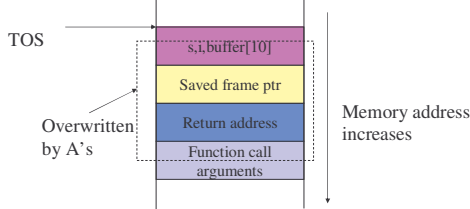

```
void sample_function(char* s)
{
    char buffer[10];
    strcpy(buffer, string);
    return;
}
```
 - Ø And a main program


```
void main()
{
    int i;
    char buffer[200];
    for(i=0; i<200;i++) buffer[i]='A';
    sample_function(buffer);
    return;
}
```
- Argument is larger than we expected

40

Stack-based Overflow Attacks

- Ø Large input will be stored on the stack, overwriting information



41

Stack-based Overflow Attacks

- Ø Attacker overwrites return address to point somewhere else
- Ø 'Local variables' portion of the stack
- Ø Places attack code in machine language at that portion
- Ø Since it is difficult to know exact address of the portion, pads attack code with NOPs before and after

42

Stack-based Overflow Attacks

- Ø IDS could look for sequence of NOPs to spot buffer overflows
- Ø Attacker uses polymorphism: he transforms the code so that NOP is changed into some other command that does the same thing, e.g. MV R1, R1
- Ø Attacker XORs important commands with a key
- Ø Attacker places XOR command and the key just before the encrypted attack code, for decryption
- Ø XOR command is also obscured

43

Stack-based Overflow Attacks

- Ø What type of commands does the attacker execute?
- Ø Commands that help him gain access to the machine
- Ø Writes a string into inetd.conf file to start shell application listening on a port, then uses Netcat to make raw interactive connection to the port
- Ø Starts TFTP to transfer Netcat onto the victim, then accesses it
- Ø Starts Xterm

44

Stack-based Overflow Attacks

- Ø How does an attacker discover stack-based overflow?
- Ø Looks at the source code
- Ø Runs application on his machine, tries to supply long inputs and looks at system registers
- Ø Read more at
 - Ø <http://packetstormsecurity.nl/docs/hack/smashstack.txt>

45

Defenses Against Stack-based Overflow

- Ø For system administrators:
 - Ø Apply patches, keep systems up-to-date
 - Ø Disable execution from the stack
 - Ø Monitor writes on the stack
 - Ø Store return address somewhere else
 - Ø Monitor outgoing traffic
- Ø For software designers
 - Ø Apply checks for buffer overflows
 - Ø Use safe functions

46

Password Attacks

- Ø Attacker attempts to login with some known username, and to guess a password
 - Ø Trying dictionary words
 - Ø Trying combinations of dictionary words
 - Ø Performing brute-force search
- Ø Attacker steals encrypted or hashed password file and tries to decrypt it

47

Defenses Against Password Attacks

- Ø Make strong passwords
 - Ø Think of a phrase, take first letters, mix big caps and special characters
 - Ø Use password filtering software
 - Ø Use strong encryption/hash techniques

48

Web Application Attacks

- Ø Account harvesting
 - Ø Gather usernames by observing error messages, then try to guess passwords
 - Ø Defense: use same error messages for everything
- Ø Hijack a session ID
 - Ø Observe session ID and how it changes between sessions
 - Ø Change your session ID to another one
 - Ø Defense: digitally sign or hash session ID, make them long enough and apply timestamps

49

Web Application Attacks

- Ø SQL Piggybacking
 - Ø Malformed input into Web form may trigger informative message from an SQL server
Input: 111111111'
Error in SQL syntax near 111111111' at line 1
SELECT * FROM account WHERE (userid='10001' and number='111111111'")
 - Ø Attacker then adds SQL commands into input
Input: 111111111'+or+userid%3d'10002
SELECT * FROM account WHERE (userid='10001' and number='111111111' or userid='10002')
 - Ø Defense: filter user input

50