

# 1. Network Security: Introduction

Simona Buchovecká, Tomáš Čejka

Faculty of Information Technology, CTU in Prague

*simona.buchovecka@fit.cvut.cz, tomas.cejka@fit.cvut.cz*

September 20, 2020

## Section 1

### About Teachers

## Ing. Tomáš Čejka, Ph.D.

- Scope of interest: network monitoring, anomaly detection, network security
- Studied at CTU in Prague, FEE (Bc.), FIT (Ing., Ph.D.)
- Researcher&Developer in CESNET  
Leader of a research&development team
- Participant on several projects related to network monitoring and network security
- Supervisor of many (successful) bachelor/master thesis
- Leader of a research&development team here at FIT CTU in Prague
  - Network Traffic Monitoring Laboratory
  - <https://netmon.fit.cvut.cz>

# Contact

## Tomas Cejka

- cejkato2@fit.cvut.cz
- cejkat@cesnet.cz
- @tomcejka
- A-954

PGP (cejkato2):

1F46 1E9E 7248 99FB 2666 8E6B 512E 05B5 D9B5 0B1B

PGP (cejkat):

BB66 06E7 08E9 836D 3936 B5B2 8F63 32E3 D255 DA7A

## Ing. Simona Buchovecká

- Ph.D. candidate
- Studied at CTU in Prague, FEE (Bc.), FIT (Ing.)
- Cyber & Privacy - Threat Management Leader at PwC
- Teacher of English course (MIE-SIB)

## Section 2

### Course Introduction

# “Disclaimer”

- Attacking someone or someone's device(s) is BAD — don't do it!!!  
(without prior written agreement)
- IT Crowd Piracy warning:  
<https://www.youtube.com/watch?v=ALZZx1xmAzg>
- Defense is very important — we should learn how attacks work in order to prevent them.
- Attacks are very frequent — we must be prepared.

# Rules of This Course

- See Course Pages: cs / en
- Homeworks
- Tutorials

## Section 3

### Insight to Network Security

# Event & Incident (NIST framework)

- **Event** is any observable occurrence in a system or network
- **Adverse events** are events with a negative consequence, such as system crashes, packet floods, unauthorized use of system privileges, unauthorized access to sensitive data, and execution of malware that destroys data
- **A computer security incident** is a violation or imminent threat of violation of computer security policies, acceptable use policies, or standard security practices.
- Security Incident != Operations incident (different objectives)

# (Network) Security Mission



## Data:

- At rest
- In transit

# Security & Risk Management

- Security is aimed at preventing loss or disclosure of data, while sustaining authorized access
- Risk = threat\*vulnerability
- Security aims to remove vulnerabilities and blocking threat agents/events
- Risk management
  - Identifying factors that could damage or disclose data
  - Evaluating those factors – data value vs. countermeasure cost
  - Security
  - Implementing cost effective countermeasures

# Common Methods to Mitigate Risks

- Compartmentalize
- Secure Fail
- Defense-in-Depth
- Least privilege
- Security-by-Obscurity

# The Weakest Link of (Network) Security

- (In)Secure protocols?
- Passwords?
- Client certificates?

**NO!**

- The **human factor** is the weakest link
- Kevin Mitnick: "... I could often get passwords and other pieces of sensitive information from companies by pretending to be someone else and JUST ASK FOR IT"
- Important detail: attacker must pretend to be an insider

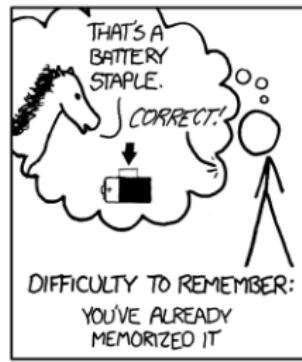
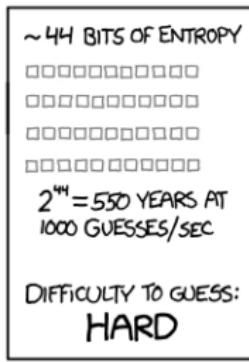
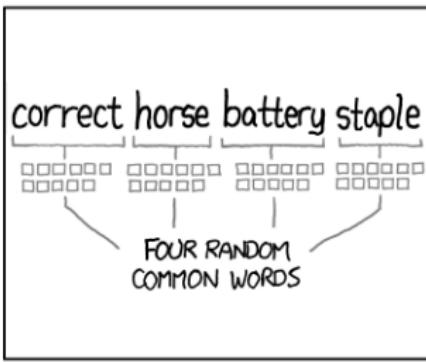
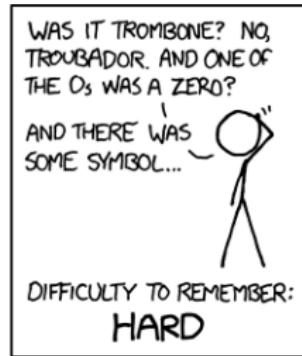
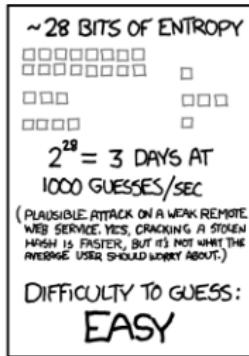
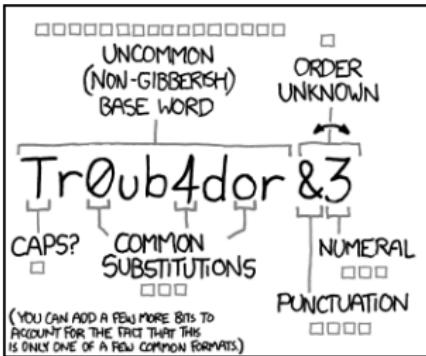
# Weakest Links of Network Security

## The Human Factor:

- The trust of humans can be manipulated by social engineers
- No matter how advanced technological security measures

## Protocol and service related weaknesses:

- Authentication: fake IP or MAC addresses, etc.
- Authorization: fake servers like DHCP, DNS, etc.



THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

<https://xkcd.com/936/>

## Section 4

### Network Models and Protocols

# Open Systems Interconnection Model

	Layer	Function	Data Unit
Host Layers	7. Application	Network process ↔ Application	Data
	6. Presentation	Data representation Data encryption/decryption Machine dependent ↔ independent	
	5. Session	Interhost communication	
	4. Transport	End-to-end connections & reliability, Flow control	Segment
	3. Network	Path determination, Logical addressing	Packet
	2. Data Link	Physical addressing	Frame
Media Layers	1. Physical	Transmission (media, signal, binary)	bit

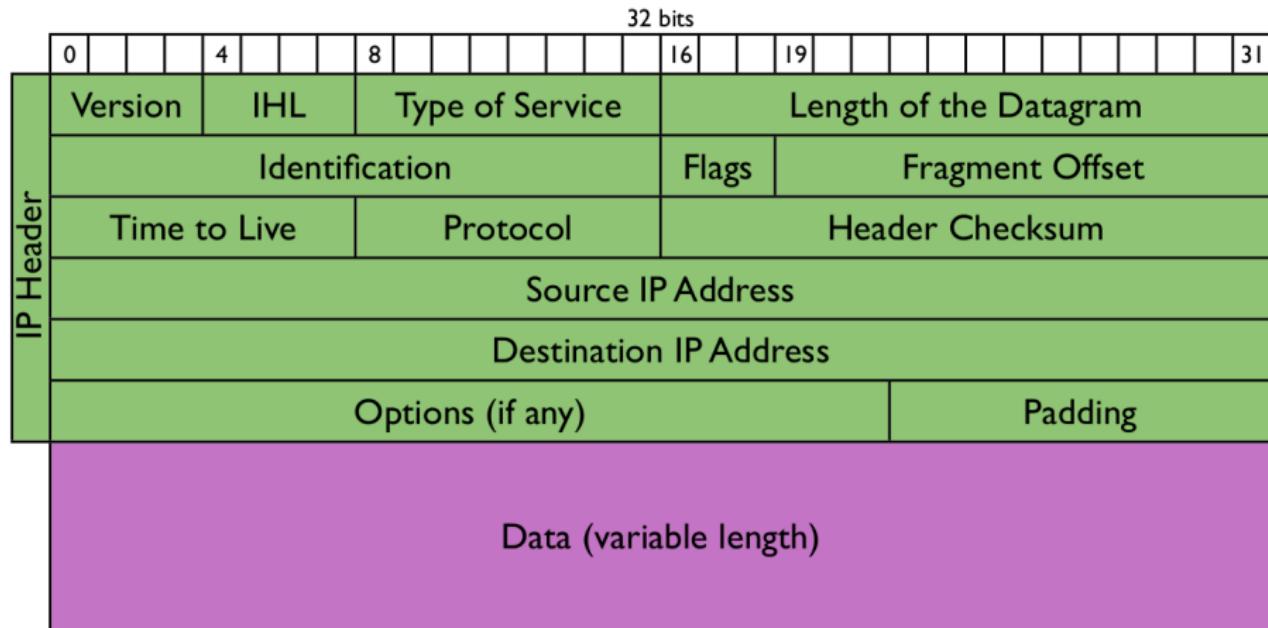
Text Source: Wikipedia.org

# Internet Protocol Suite - RFC 1122

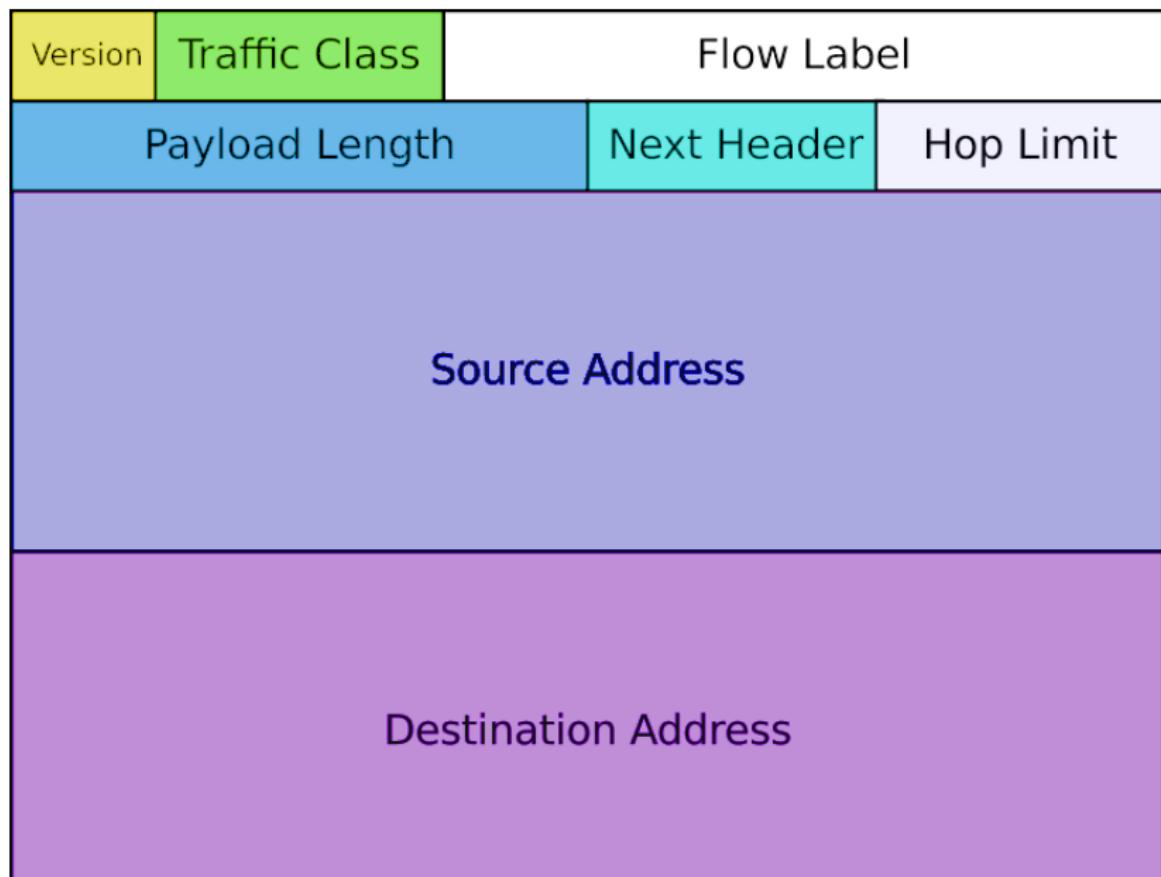
Text Source: Wikipedia.org

Layer	Protocols	Our Focus	
4. Application	DHCP, DNS, TFTP, TLS/SSL, FTP, Gopher, HTTP, IMAP, IRC, NNTP,	DHCP (DNS MiM Attack)	Firewalls
	Routing protocols like BGP and RIP which run over TCP/UDP	Encryption Authorization	
3. Transport	TCP, UDP, DCCP, SCTP, IL, RUDP, RSVP		
2. Internet	IP (IPv4, IPv6), ICMP, IGMP, ICMPv6	NAT	Firewalls
	OSPF for IPv4 – has been moved to the Link layer since RFC 2740		
I. Link	ARP, RARP, OSPF (IPv4/IPv6), ISIS, NDP	ARP MiM Attack	

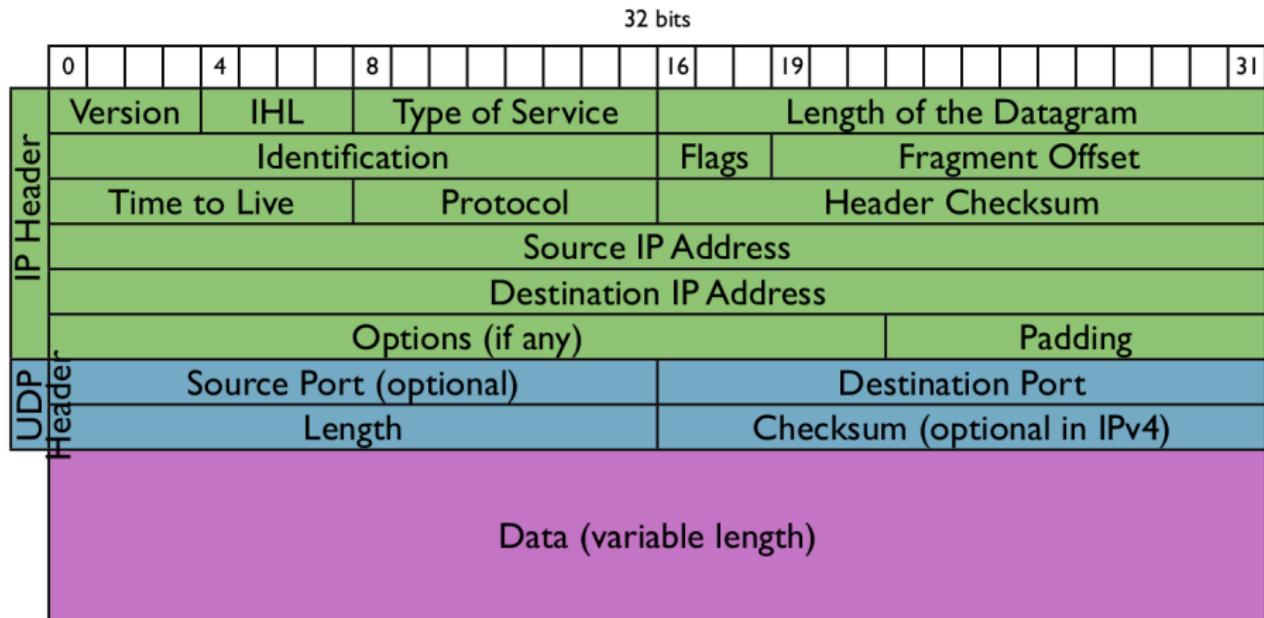
# IPv4 Header Structure



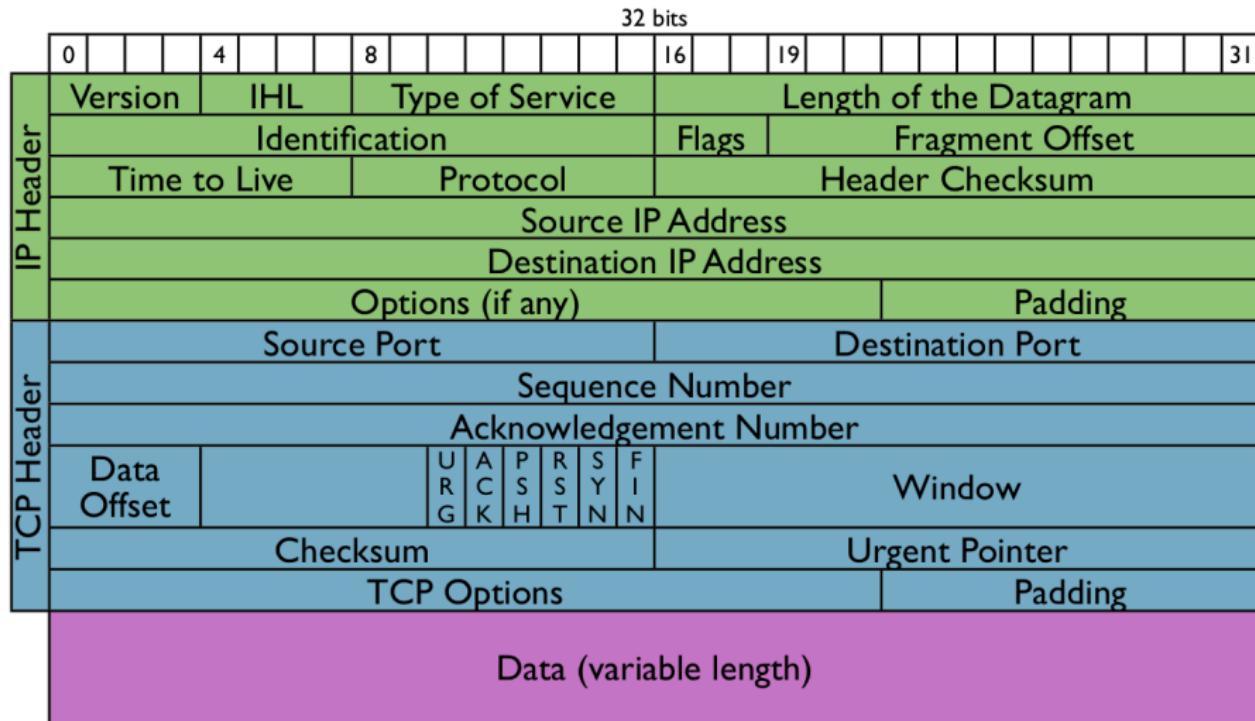
# IPv6 Header Structure



# UDP Header Structure



# TCP/IP Packet Structure



# ARP Header Structure

**Internet Protocol (IPv4) over Ethernet ARP packet**

<b>octet offset</b>	<b>0</b>	<b>1</b>
<b>0</b>	Hardware type (HTYPE)	
<b>2</b>	Protocol type (PTYPE)	
<b>4</b>	Hardware address length (HLEN)	Protocol address length (PLEN)
<b>6</b>	Operation (OPER)	
<b>8</b>	Sender hardware address (SHA) (first 2 bytes)	
<b>10</b>	(next 2 bytes)	
<b>12</b>	(last 2 bytes)	
<b>14</b>	Sender protocol address (SPA) (first 2 bytes)	
<b>16</b>	(last 2 bytes)	
<b>18</b>	Target hardware address (THA) (first 2 bytes)	
<b>20</b>	(next 2 bytes)	
<b>22</b>	(last 2 bytes)	
<b>24</b>	Target protocol address (TPA) (first 2 bytes)	
<b>26</b>	(last 2 bytes)	

# DHCP

Dynamic Host Configuration Protocol

**Allows a computer in a LAN to be configured automatically:**

- IP Address
- Gateway
- DNS Servers etc...

Maintains a database for keeping track of connected computers

# Operation Phases: DHCP Discovery

- The client broadcasts messages on the physical subnet to
  - discover available DHCP servers
  - User Datagram Protocol (UDP) packet
  - with the broadcast destination 255.255.255.255 (or a specific subnet broadcast address)

# Operation Phases: DHCP Offer

- A DHCP server receives an IP lease request
- Reserves an IP address for the client
- Sends a DHCPOFFER message to the client
- The message contains:
  - The client's MAC address
  - the offered IP address
  - a subnet mask
  - the lease duration
  - and the IP address of the DHCP server

# Operation Phases: DHCP Acknowledgement

- The server sends the client a DHCPPACK packet with:
  - the lease duration
  - any other configuration information the client requested.
- This completes the IP configuration process

# Operation Phases: DHCP Attacks

- Two types of attacks
  - Unauthorized DHCP Servers (Rogue Servers)
  - Falsified DHCP Clients (DHCP Starvation)
- Rogue DHCP Server
  - A trojan installed on an infected machine
  - Serves bogus DHCP packets to other machines
  - If the Trojan is fast it can modify the network configuration of other computers.
- DHCP Starvation
  - Use-up IP Addresses

## Section 5

### Basic Defense: Packet Filtering

# Firewalls

- Block unauthorized access
- Permit authorized communications
- Often provide NAT and DHCP
  - Example: Basic residential routers
- Software firewalls can be installed on a host to protect a single computer

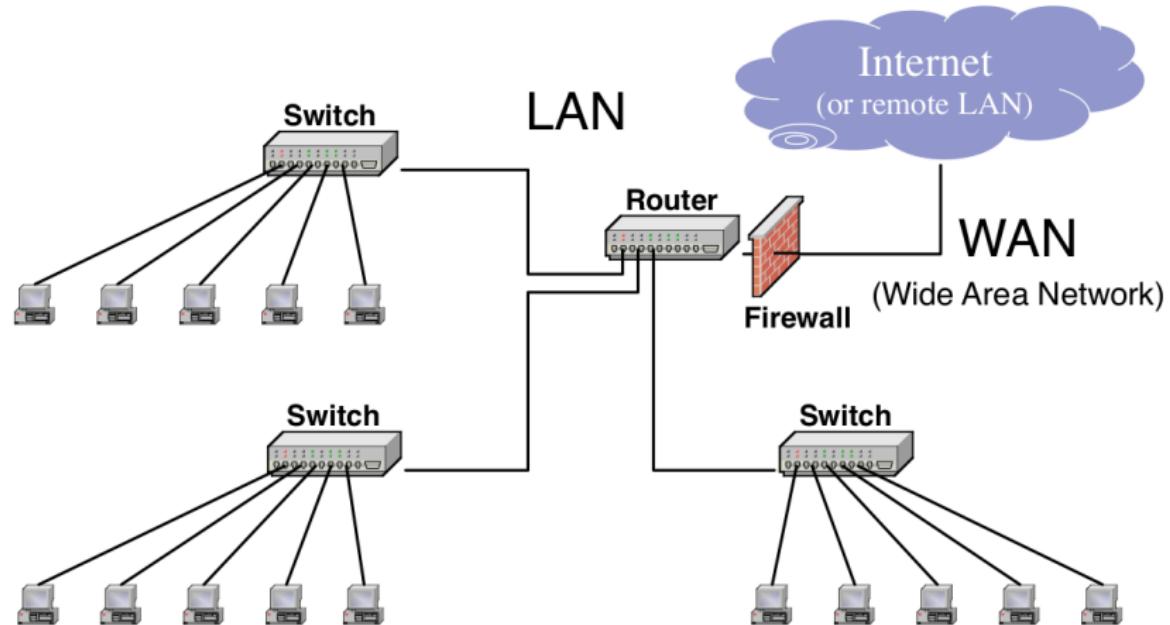
# Types of Firewalls

- Packet filter: inspects each packet and apply specified rules
- Application layer: “understand” certain applications and protocols (FTP, DNS, web)
- Stateful filter: maintain sessions or network flows to detect out-of-place packets
- NAT: Provides basic firewalling protection

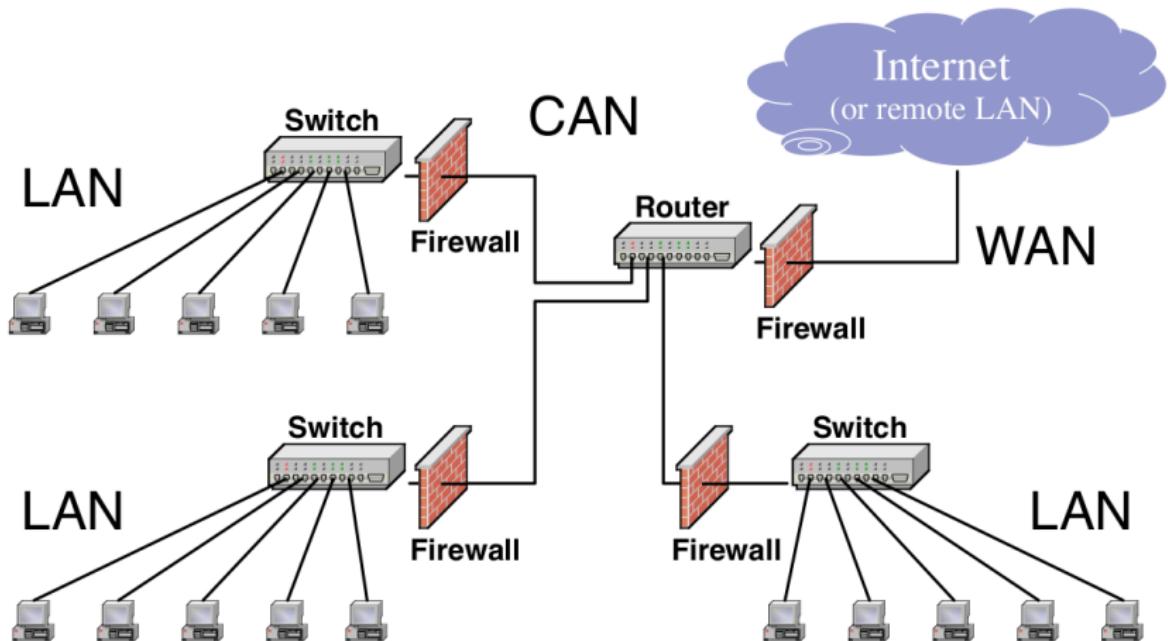
## Practical self-study:

investigate *iptable*, *nftables*, *firewalld*

# LAN Security #1



# LAN Security #2



# Questions?

# Network Attacks Insight/Overview

Simona Buchovecká, Tomáš Čejka

Faculty of Information Technology, CTU in Prague

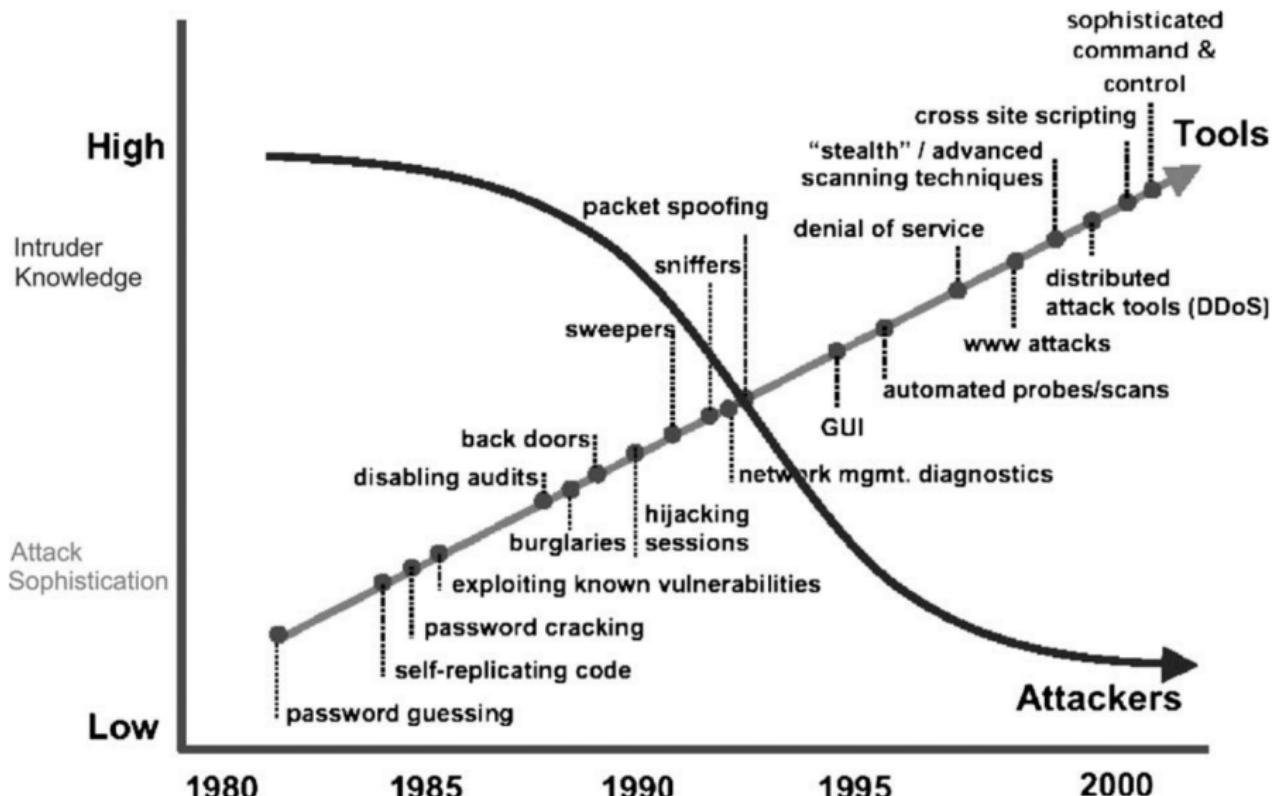
*simona.buchovecka@fit.cvut.cz, tomas.cejka@fit.cvut.cz*

February 21, 2021

## Section 1

### Evolution

# History of Attacks



# Present, Discussion

- Ransomware
- Wireless attacks (KRACK)
- IoT botnets (Mirai)
- Philips hue
- Cryptominers
- Hardware attacks  
(CPU arch: Spectre, meltdown, cache attacks)
- “side-channel attacks”

# Botnets

- Infected devices, synchronized, collaborating
- Different ways of communication:
  - Central Command&Control (C&C / C2) servers (channels: IRC, ICQ, HTTP, favicon, DNS)
  - P2P botnets
- Usually fastflux domains

## Section 2

### Classification

# Attack Vector / Indicator of Compromise

**Attack Vector** describe how an attack can be performed and what it exploits.

**Indicator of compromise** in computer forensics is an artifact observed on a network or in an operating system that with high confidence indicates a computer intrusion.

**Examples:** Fragmented packets exploiting buffer overflow vulnerability in some particular software; packets with spoofed srcip with 123/UDP dstport sent to an NTP server.

# Ways of Classification

There are many different classification methodologies.

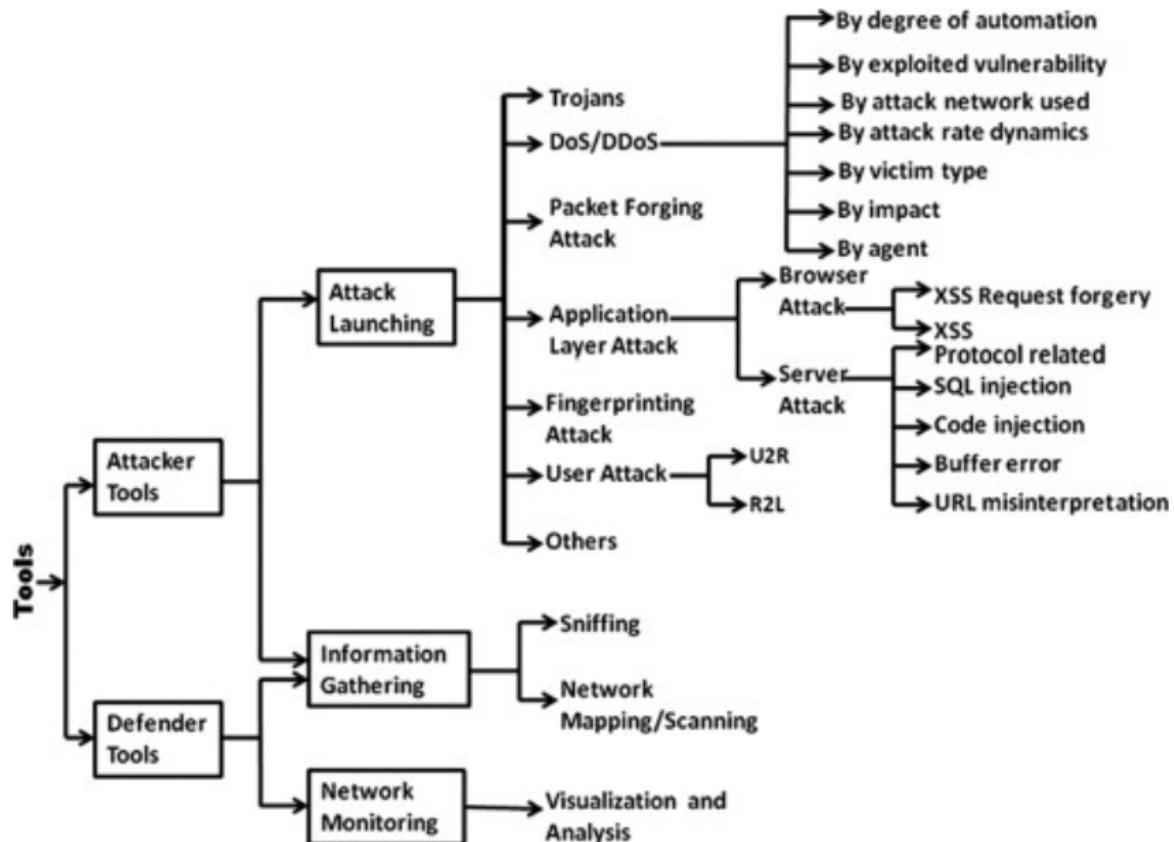
Hansman et al.: Based on dimensions:

- 1st dimension to categorise the attack based on attack vector,
- 2nd dimension based on attack targets,
- 3rd dimension covers vulnerabilities and exploits that attack uses,
- 4th dimension deals with attacks having payloads or effects beyond themselves,
- other dimensions can be added.

There are many taxonomies of attack techniques, e.g.,

<https://attack.mitre.org/> is popular.

# Example of taxonomy



# Brief List of Attack Types

- Information Gathering:
  - Scanning (vertical/horizontal)
  - OSINT (Open Source Intelligence), \*INT
- Credential Stealing
  - Phishing
  - Brute-force attacks (dictionary attacks)
- Communication intercept
  - Man-in-the-Middle
  - Poisoning
  - Hijacking
- Service/operation disruption
  - (D)DoS
  - Starvation
  - De-authentication/Connection resetting
- Data Exfiltration
  - Covert Channels
  - Tunnels / VPNs

## Section 3

Related Topics

# Forms of Protection

- Access Control
- Authentication
- Confidentiality
- Integrity
- Non-repudiation

# Sources of Security Threats

- Design Philosophy
- Weaknesses in Network Infrastructure and Communication Protocols
- Rapid Growth of Cyberspace
- The Growth of the Hacker Community
- Vulnerability in Operating System Protocol
- The Invisible Security Threat: The Insider Effect
- Social Engineering
- Physical Theft

# Security Threat Motives

- Terrorism
- Military Espionage
- Economic Espionage
- Targeting the National Information Infrastructure
- Vendetta/Revenge
- Hate (National Origin, Gender, and Race)
- Notoriety
- Greed
- Ignorance

## Section 4

### Observation & Monitoring

# Attack Observation via Monitoring

## General Classification

- Host-Based (system logs, auditing tools, . . . )
- Network-Based

## Interaction in the network

- Active (ping, iperf, traceroute, Atlas RIPE, PerfSonar)
- Passive

# Attack Observation via Monitoring

## Monitoring data unit

- Counter
  - High-level information (total numbers of packets/bytes/errors, packet loss)
  - e.g. SNMP, Network Telemetry
- Packet
  - “Raw data”
  - Deep Packet Inspection (DPI)
  - Pattern matching
- Flow
  - high-level overview, communication of devices without full content
  - aggregation

# IP Flow

*An IP Flow, also called a Flow, is defined as a set of IP packets passing an Observation Point in the network during a certain time interval. All packets that belong to a particular Flow have a set of common properties derived from the data contained in the packet and from the packet treatment at the Observation Point.*

(Cisco Systems NetFlow Services Export Version 9)

# Classification of IP Flows

- uni-flow
  - unidirectional communication between srcip and dstip
- bi-flow
  - bidirectional
  - pairing flow records in time
  - advantage: requests and responses are matched before analysis

# How Do Attacks Look Like?

Packet point of view

1	0.000000	172.16.0.8	64.13.134.52	TCP	58	36050	443	36050 - 443 [SYN] Seq=0 Win=3072 Len=0 MSS=1460
2	0.001539	172.16.0.8	64.13.134.52	TCP	58	36050	143	36050 - 143 [SYN] Seq=0 Win=3072 Len=0 MSS=1460
3	0.001597	172.16.0.8	64.13.134.52	TCP	58	36050	3306	36050 - 3306 [SYN] Seq=0 Win=2048 Len=0 MSS=1460
4	0.001650	172.16.0.8	64.13.134.52	TCP	58	36050	199	36050 - 199 [SYN] Seq=0 Win=3072 Len=0 MSS=1460
5	0.001703	172.16.0.8	64.13.134.52	TCP	58	36050	111	36050 - 111 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
6	0.001755	172.16.0.8	64.13.134.52	TCP	58	36050	1025	36050 - 1025 [SYN] Seq=0 Win=4096 Len=0 MSS=1460
7	0.001807	172.16.0.8	64.13.134.52	TCP	58	36050	995	36050 - 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
8	0.001861	172.16.0.8	64.13.134.52	TCP	58	36050	587	36050 - 587 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
9	0.001913	172.16.0.8	64.13.134.52	TCP	58	36050	53	36050 - 53 [SYN] Seq=0 Win=3072 Len=0 MSS=1460
10	0.001965	172.16.0.8	64.13.134.52	TCP	58	36050	5900	36050 - 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
11	0.063797	64.13.134.52	172.16.0.8	TCP	60	53	36050	53 - 36050 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1380
12	0.065271	172.16.0.8	64.13.134.52	TCP	58	36050	21	36050 - 21 [SYN] Seq=0 Win=4096 Len=0 MSS=1460
13	0.065341	172.16.0.8	64.13.134.52	TCP	58	36050	113	36050 - 113 [SYN] Seq=0 Win=4096 Len=0 MSS=1460
14	0.126832	64.13.134.52	172.16.0.8	TCP	60	113	36050	113 - 36050 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
15	0.129000	172.16.0.8	64.13.134.52	TCP	58	36050	80	36050 - 80 [SYN] Seq=0 Win=3072 Len=0 MSS=1460
16	0.129075	172.16.0.8	64.13.134.52	TCP	58	36050	139	36050 - 139 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
17	0.189975	64.13.134.52	172.16.0.8	TCP	60	80	36050	80 - 36050 [SYN, ACK] Seq=0 Ack=1 Win=15840 Len=0 MSS=1380
18	0.191518	172.16.0.8	64.13.134.52	TCP	58	36050	3389	36050 - 3389 [SYN] Seq=0 Win=3072 Len=0 MSS=1460
19	0.191589	172.16.0.8	64.13.134.52	TCP	58	36050	23	36050 - 23 [SYN] Seq=0 Win=2048 Len=0 MSS=1460
20	1.202878	172.16.0.8	64.13.134.52	TCP	58	36051	23	36051 - 23 [SYN] Seq=0 Win=2048 Len=0 MSS=1460
21	1.202974	172.16.0.8	64.13.134.52	TCP	58	36051	3389	36051 - 3389 [SYN] Seq=0 Win=2048 Len=0 MSS=1460
22	1.203041	172.16.0.8	64.13.134.52	TCP	58	36051	139	36051 - 139 [SYN] Seq=0 Win=4096 Len=0 MSS=1460
23	1.203111	172.16.0.8	64.13.134.52	TCP	58	36051	21	36051 - 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
24	1.203176	172.16.0.8	64.13.134.52	TCP	58	36051	5900	36051 - 5900 [SYN] Seq=0 Win=3072 Len=0 MSS=1460
25	1.203241	172.16.0.8	64.13.134.52	TCP	58	36051	587	36051 - 587 [SYN] Seq=0 Win=4096 Len=0 MSS=1460
26	1.203316	172.16.0.8	64.13.134.52	TCP	58	36051	995	36051 - 995 [SYN] Seq=0 Win=2048 Len=0 MSS=1460
27	1.203381	172.16.0.8	64.13.134.52	TCP	58	36051	1025	36051 - 1025 [SYN] Seq=0 Win=3072 Len=0 MSS=1460
28	1.203446	172.16.0.8	64.13.134.52	TCP	58	36051	111	36051 - 111 [SYN] Seq=0 Win=4096 Len=0 MSS=1460
29	1.203514	172.16.0.8	64.13.134.52	TCP	58	36051	199	36051 - 199 [SYN] Seq=0 Win=3072 Len=0 MSS=1460
30	1.203581	172.16.0.8	64.13.134.52	TCP	58	36051	3306	36051 - 3306 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
31	1.203651	172.16.0.8	64.13.134.52	TCP	58	36051	143	36051 - 143 [SYN] Seq=0 Win=4096 Len=0 MSS=1460
32	1.203716	172.16.0.8	64.13.134.52	TCP	58	36051	443	36051 - 443 [SYN] Seq=0 Win=2048 Len=0 MSS=1460
33	1.402807	172.16.0.8	64.13.134.52	TCP	58	36050	1723	36050 - 1723 [SYN] Seq=0 Win=4096 Len=0 MSS=1460
34	1.402891	172.16.0.8	64.13.134.52	TCP	58	36050	993	36050 - 993 [SYN] Seq=0 Win=4096 Len=0 MSS=1460
35	1.402958	172.16.0.8	64.13.134.52	TCP	58	36050	110	36050 - 110 [SYN] Seq=0 Win=4096 Len=0 MSS=1460
36	1.403023	172.16.0.8	64.13.134.52	TCP	58	36050	8080	36050 - 8080 [SYN] Seq=0 Win=4096 Len=0 MSS=1460
37	1.403088	172.16.0.8	64.13.134.52	TCP	58	36050	1720	36050 - 1720 [SYN] Seq=0 Win=4096 Len=0 MSS=1460

(wireshark)

# Some Abbreviations

*srcip* Source IP

*dstip* Destination IP

*srcport* Source port of transport protocol

*dstport* Destination port of transport protocol

*proto* Transport protocol (according to *proto* field in Network protocol header)

# How Do “Attacks” Look Like?

Flow point of view

TIMEFIRST	TIMELAST	SRCIP:SRCPORT->DSTIP:DSTPORT	PROTO	FLG	PKTS	#B
8:09	8:09	46.28.11.24:123 -> 10.0.1.15:42958	UDP	.....	1	76
8:09	8:09	10.0.1.15:42958 -> 46.28.11.24:123	UDP	.....	1	76
8:09	8:09	0.0.0.0:0 -> 224.0.0.1:0		2	.....	1 32
8:09	8:09	10.0.1.1:53 -> 10.0.1.15:46187	UDP	.....	2	344
8:10	8:10	10.0.1.1:0 -> 10.0.1.15:0	ICMP	.....	199	208854
8:10	8:10	10.0.1.15:46501 -> 10.0.1.1:53	UDP	.....	2	126
8:10	8:10	10.0.1.15:0 -> 10.0.1.1:2048	ICMP	.....	199	208854
8:10	8:10	10.0.1.15:50645 -> 10.0.1.1:53	UDP	.....	2	124
8:10	8:10	10.0.1.1:53 -> 10.0.1.15:55978	UDP	.....	2	344
8:10	8:11	10.0.1.1:0 -> 10.0.1.15:0	ICMP	.....	3096	3256202
8:10	8:11	10.0.1.15:0 -> 10.0.1.1:2048	ICMP	.....	3096	3256202
8:10	8:11	10.0.1.1:22 -> 10.0.1.15:34974	TCP	.AP...	2484	835296
8:10	8:11	10.0.1.15:34974 -> 10.0.1.1:22	TCP	.AP...	1903	99652
8:11	8:11	10.0.1.1:53 -> 10.0.1.15:56957	UDP	.....	2	242
8:09	8:12	10.0.1.220:5353 -> 224.0.0.251:5353	UDP	.....	43	6665

# How Do Attacks Look Like?

Alert point of view

```
{"Category": ["Malware"], "Node": [{"AggrWin": "00:05:00", "SW": ["Nemea", "urlblacklistfilter"], "Type": ["Flow", "Blacklist"], "Name": "cz.cesnet.nemea.urlblacklist"}], "EventTime": "2018-09-28T17:28:24Z", "Description": "URL: 'vseccz.weebly.com' (listed: Malware Domains) was requested by 146.102.131.199.", "Format": "IDEAO", "CeaseTime": "2018-09-28T17:28:41Z", "CreateTime": "2018-09-28T17:30:58Z", "Note": "URL: 'vseccz.weebly.com' was found on blacklist(s): Malware Domains.", "Source": [{"InFlowCount": 4, "Proto": ["tcp"], "Hostname": "vseccz.weebly.com", "InByteCount": 3136, "InPacketsCount": 30, "IP4": ["199.34.228.53"], "Type": ["OriginBlacklist"], "Port": [443]}, {"IP4": ["146.102.131.199"], "Proto": ["tcp"]}], "DetectTime": "2018-09-28T17:28:41Z", "Ref": ["http://mirror1.malwaredomains.com/files/justdomains"], "ID": "aaf1206b-f7e7-419a-898b-72447a1ed72c"}
```

# How Do Attacks Look Like?

Alert point of view

RealTime Events								
ST	CNT	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	2	2017-02-11 03:02:41	10.3.14.134	51734	10.3.14.2	53	17	ET DNS Query to a *.top domain - Likely Hostile
RT	4	2017-02-11 03:02:43	10.3.14.134	49249	104.155.4.180	80	6	ET INFO HTTP Request to a *.top domain
RT	1	2017-02-11 03:02:43	104.155.4.180	80	10.3.14.134	49249	6	ET POLICY PE EXE or DLL Windows file download
RT	1	2017-02-11 03:02:43	104.155.4.180	80	10.3.14.134	49249	6	ET POLICY Binary Download Smaller than 1 MB Likely Hostile
RT	2	2017-02-11 03:02:43	104.155.4.180	80	10.3.14.134	49249	6	ET CURRENT_EVENTS Likely Evil EXE download from MSXMLHTTP non-exe extension M2
RT	2	2017-02-11 03:02:43	104.155.4.180	80	10.3.14.134	49249	6	ET TROJAN JS/WSF Downloader Dec 08 2016 M6
RT	2	2017-02-11 03:02:43	104.155.4.180	80	10.3.14.134	49249	6	ET INFO Possible EXE Download From Suspicious TLD
RT	2	2017-02-11 03:02:43	104.155.4.180	80	10.3.14.134	49249	6	ET INFO EXE - Served Attached HTTP
RT	1	2017-02-11 03:02:44	10.3.14.134	51735	91.119.56.0	6892	17	ET TROJAN Ransomware/Cerber Checkin M3(4)
RT	1	2017-02-11 03:02:44	10.3.14.134	51735	91.121.56.30	6892	17	ET TROJAN Possible Downadup/Conficker-C P2P encrypted traffic UDP Ping Packet (bit ...)
RT	1	2017-02-11 03:02:50	10.3.14.134	51736	91.119.56.0	6892	17	ET TROJAN W32/Cerber.Ransomware CnC Checkin M4
RT	1	2017-02-11 03:02:54	10.3.14.134	49250	54.87.5.88	80	6	ETPRO TROJAN Cerber Blockchain Query
RT	2	2017-02-11 03:02:54	10.3.14.134	50205	10.3.14.2	53	17	ET TROJAN Ransomware/Cerber Onion Domain Lookup
RT	7	2017-02-11 03:03:21	67.210.245.241	80	10.3.14.131	49506	6	ET SHELLCODE UTF-8/16 Encoded Shellcode
RT	2	2017-02-11 03:03:21	67.210.245.241	80	10.3.14.131	49506	6	ET WEB_CLIENT Possible String.FromCharCode Javascript Obfuscation Attempt
RT	1	2017-02-11 03:04:07	10.3.14.131	49585	54.229.205.204	12080	6	ET POLICY HTTP Request on Unusual Port Possibly Hostile
RT	1	2017-02-11 03:04:07	10.3.14.131	49585	54.229.205.204	12080	6	ET POLICY HTTP POST on unusual Port Possibly Hostile
RT	2	2017-02-11 03:05:47	10.3.14.131	64890	10.3.14.2	53	17	ET TROJAN Spora Ransomware DNS Query

# How Do Attacks Look Like?

Incident point of view in IODEFv2 — ex. C2 domains from a given campaign

<https://tools.ietf.org/html/rfc7970#section-7.2>

```
...
<IndicatorData>
  <Indicator>
    <IndicatorID name="csirt.example.com" version="1">G90823490</
      IndicatorID>
    <Description>C2 domains </Description>
    <StartTime>2014-12-02T11:18:00-05:00</StartTime>
    <Observable>
      <BulkObservable type="fqdn">
        <BulkObservableList>
          kj290023j09r34.example.com
          09ijk23jfj0k8.example.net
          klknjwfjiowjefr923.example.org
          oimireik79msd.example.org
        </BulkObservableList>
      </BulkObservable>
    </Observable>
  </Indicator>
</IndicatorData>
...
```

## Section 5

Closing Words

## (Recommended) Resources

- Simon Hansman, Ray Hunt: *A taxonomy of network and computer attacks*, 2005, <https://doi.org/10.1016/j.cose.2004.06.011>.
- N. Hoque, Monowar H. Bhuyan, R.C. Baishya, D.K. Bhattacharyya, J.K. Kalita: *Network attacks: Taxonomy, tools and systems*, 2014, <https://doi.org/10.1016/j.jnca.2013.08.001>.

# Questions?

## 3. Network Attacks and Their Detection

### (D)DoS, Scanning, Brute-Force

Simona Buchovecká, Tomáš Čejka

Faculty of Information Technology, CTU in Prague

*simona.buchovecka@fit.cvut.cz, tomas.cejka@fit.cvut.cz*

October 18, 2020

# Section 1

## General Information

# Terminology

*srcip* Source IP

*dstip* Destination IP

*srcport* Source port of transport protocol

*dstport* Destination port of transport protocol

**TCP** Transmission Control Protocol

**UDP** User Datagram Protocol

## Section 2

### Scanning

# Network Scanning

- Host / Service discovery
- Information gathering
- UDP (connectionless) vs. TCP (connection-oriented)
- ICMP

# Scanning Traffic: Packet view

## Host discovery by ICMP

ICMP messages with type *echo*, usually one *srcip*, changing *dstip*

## TCP SYN

TCP packets, SYN flag, usually one *srcip*, usually one *dstip*, usually many *dstport*, one or more *srcport*

Scanner analyzes response, SYN&ACK for open port.

## UDP scan

UDP packets, usually one *srcip*, usually one *dstip*, usually many *dstport*, one or more *srcport*, some payload optional.

Scanner analyzes response, packets are being duplicated, longer timeouts.

Brainstorming: block scan? RST scan? X-mass scan?

# Scanning Traffic: Flow view

## Host discovery by ICMP

$proto = 1$ ,  $srcport = 0$ ,  $dstport = \text{type and code}$

## TCP SYN

Increased number of TCP flow records, mostly with just SYN or SYN&RST flag, with one or more  $srcport$  and usually many  $dstport$ .

## UDP scan

Increased number of UDP flow records, with one or more  $srcport$  and usually many  $dstport$ .

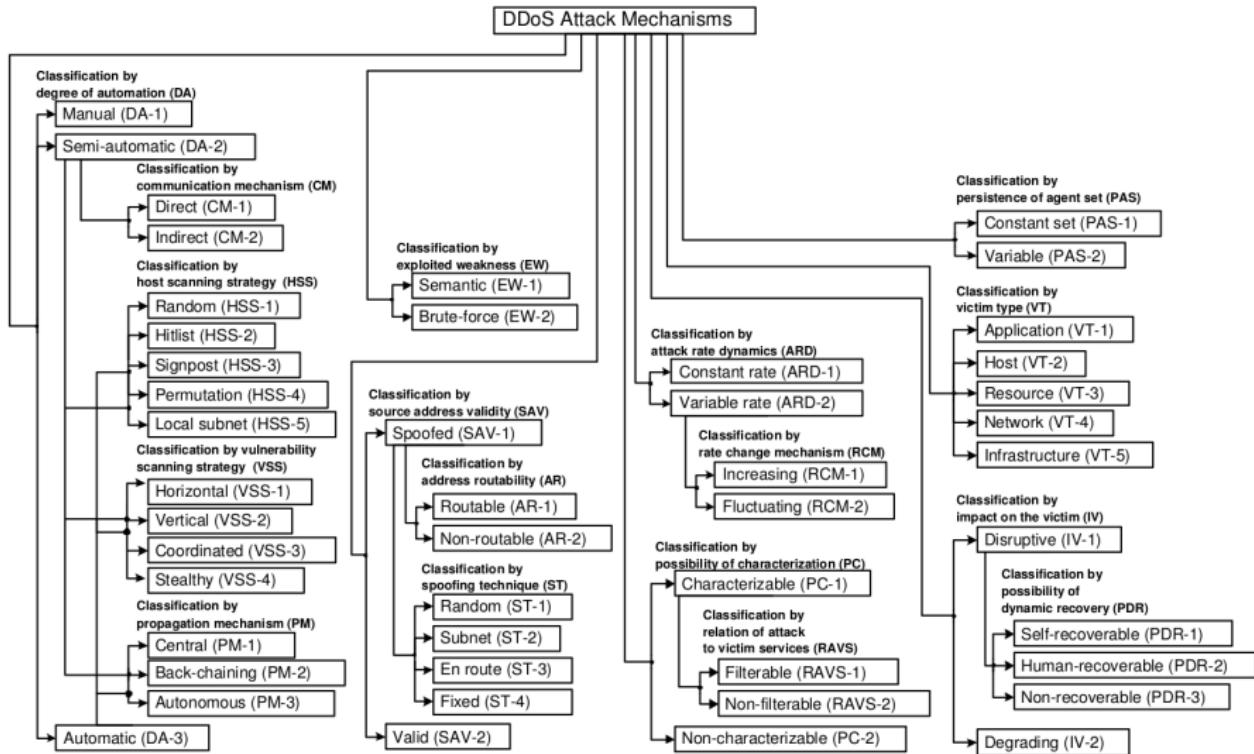
Brainstorming: block scan? RST scan? X-mass scan?

Discussion in a particular tutorial.

## Section 3

(D)DoS

# (Distributed) Denial of Service (DoS)



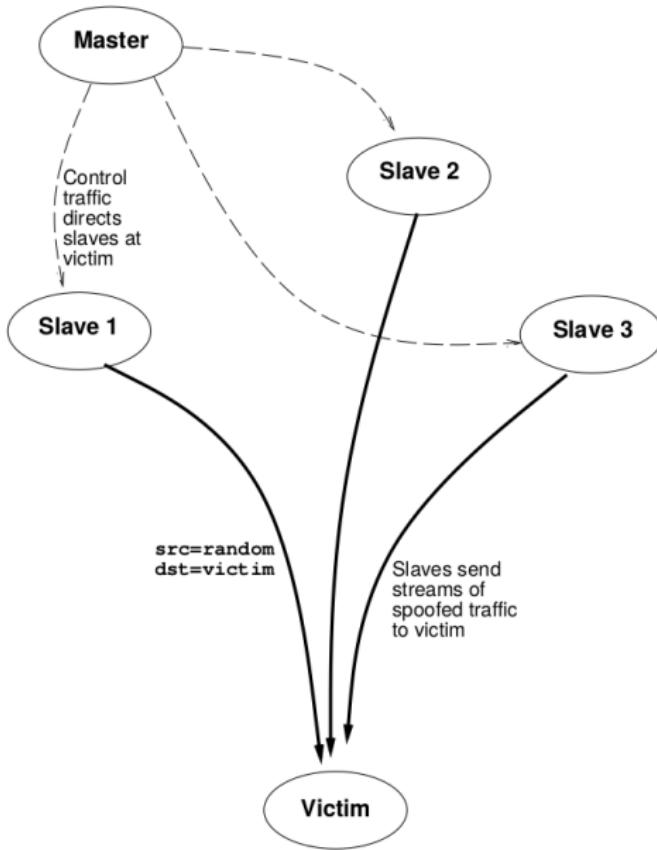
# About DoS

- Making system slow/unusable by overloading its resources (with a little computing work)
- Availability attack
- Generally: unsophisticated attack, attackers do not gain any information from the target system, BUT they can learn its defense
- There are (unfortunately) many vulnerabilities in current software/hardware
- Basically, any error that ends with crash can cause DoS
- Depletion of victim's resources can cause DoS
- Deadlocks can cause DoS
- Infinite loops can cause DoS
- Bad configuration of network infrastructure can cause DoS
- See database of Common Vulnerabilities and Exposures (CVE)  
<https://cve.mitre.org/>, <https://www.exploit-db.com/>

# “Bombs”

- Well-known fork bombs  
([https://en.wikipedia.org/wiki/Fork\\_bomb](https://en.wikipedia.org/wiki/Fork_bomb)), possible even in shell
- Attacks against parsers:
  - XDoS attack (XML) (consuming expansion:  
<https://en.wikipedia.org/wiki/BillionLaughsAttack>)
  - Multiple signatures (consuming verification)
  - Regular expression DoS (ReDoS) (consuming evaluation:  
<https://en.wikipedia.org/wiki/ReDoS>)
  - (zip) archive that is too large after extraction

# Distributed DoS (DDoS)



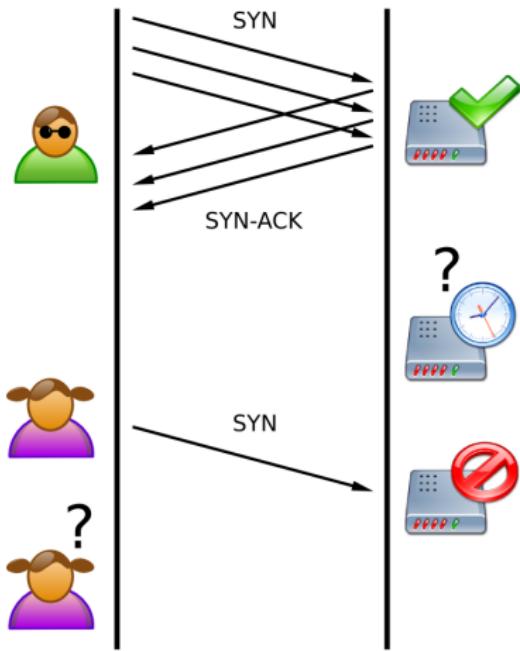
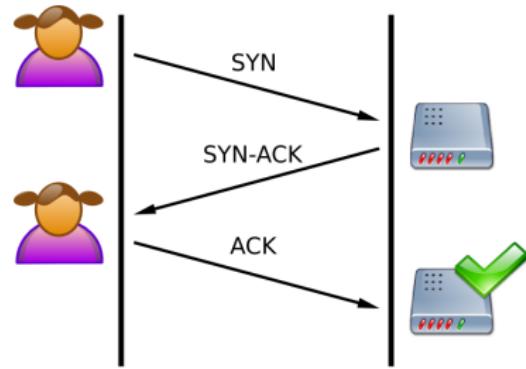
# Distributed DoS (DDoS)

- Huge (synchronized) traffic (/many requests) from many sources against a victim
- Usage of botnets, many existing, many available, famous one <https://github.com/jgamblin/Mirai-Source-Code> in 2016
- Responding honeypots

Multiple hosts affected — usually compromised

- primary victim — service under attack
- secondary victims — compromised systems launching the attack, zombies/bots (botnet)

# SYN Flood Attack



# SYN Flood Traffic

## Packets

TCP packets with SYN flag, Usually one or more *srcip*, one *dstip*, usually many *srcport*, one *dstport*.

## Flows

Increased number of TCP flow records with SYN flag, see the Packets description.

# UDP Flood Attack

- Sending large number of UDP packets to random ports
- Victim will reply with ICMP Destination unreachable packet for every UDP request to the closed ports
- Processing and sending big amount of ICMP packets may make system unresponsive for legitimate requests

# Ping of Death

- Typical size of IPv4 packet: 64 bytes
- Maximum size may be up to 65 535 bytes
- Many systems were not designed to properly process such big packets
- ICMP echo request (ping) with maximum packet size may cause buffer overflow, system instability... or other problems on the receiving/victim system

# WiFi DoS Attacks

## De-authentication attack

- IEEE 802.11 defines *deauthentication* frame (management frames are being sent unencrypted)
- AP can send the *deauthentication* frame to a station
- Attacker can send a *deauthentication* frame with a spoofed address to a victim

## Signal interference

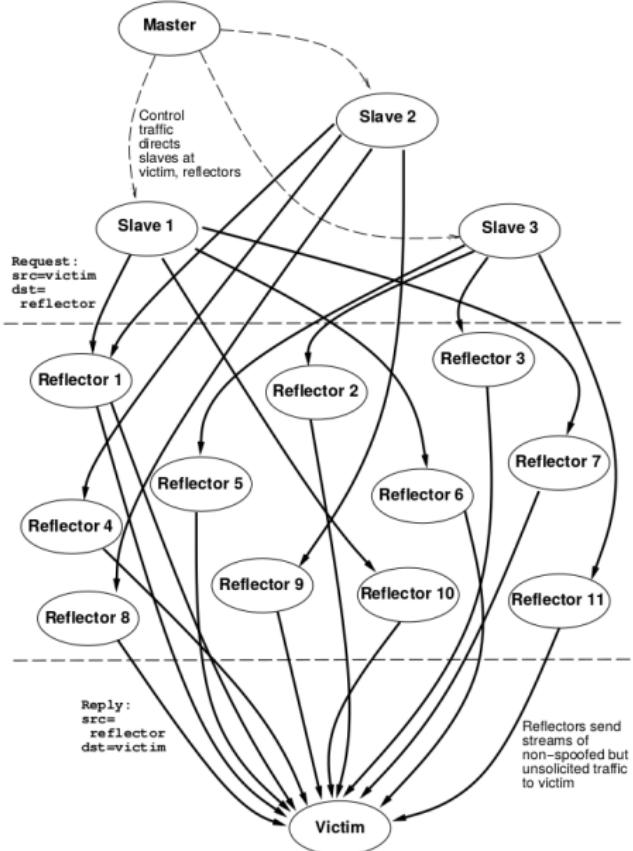
- Jamming
- WiFi jammer generates a noise on WiFi channels, making the frequencies unusable

# Application Layer (L7) DDoS

## Example: HTTP

- HTTP GET request flooding
- Volumetric attack, using a botnet to perform HTTP GET requests that pretend to be valid
- HTTP POST
- Valid POST request sent at very low rate – preventing the connection to be properly completed
- HTTP slow read
- Read the HTTP response “slowly” = set up small window (maximum amount of received data) size
- HTTP Malformed attacks
- Trying malformed/intentionally invalid requests; goal: unstable system

# Distributed Reflection DoS (DRDoS)



- Amplification often employed
  - small number of packets (bytes) from source
  - generating bigger number of packets (bytes) against destination (attack target)
- Asymmetric attack (low resources, large consequences)
- Primary victim — service under attack
- Secondary victims — compromised systems launching the attack, zombies/bots (botnet)
- Reflectors/amplifiers (e.g. open DNS resolvers)

# Smurf Attack

- reflected attack
- ICMP packets, broadcast address
- ICMP with spoofed *srcip* (victim's)

# DNS Amplification

- Amplification through DNS
- typically transmitted over UDP, i.e., *srcip* can be spoofed
- $\text{size}(\text{Response}) > \text{size}(\text{query})$
- Problem: open DNS resolvers (respond to any query by any *srcip*)
  - *srcip* spoofed with victim's address
  - small query send by attacker
  - victim will receive much larger response

# Booters Phenomenon

- Stress-test / DDoS-for-hire / DDoS-on-Demand
- Cheap attacks/stress-tests available for everyone! (It makes it more dangerous)

The screenshot shows the Netbreak website interface. At the top, there's a navigation bar with tabs like 'Home', 'Plans', 'Profile', and 'Logout'. Below the navigation is a search bar and a 'Netbreak - Account' section. The main content area is titled 'NETBREAK' and displays several service offerings:

- Subscription list:**
  - Plan - Free:** 0€/month. Methods included: UDP - DNS Any x10. 10 min seconds max. Power: 0.
  - Plan - Classic:** 9€/month. Methods included: UDP - DNS Any x10. 10 min seconds max. Power: 1.
  - Plan - Premium:** 19€/month. Methods included: UDP - DNS Any x10. 10 min seconds max. Power: 2.
  - Plan - Hard:** 45€/month. Methods included: UDP - DNS Any x10. 10 min seconds max. Power: 5.
- LoL-Dropper offers:**
  - LoL Dropper (lite):** 1€/100 seconds of drop. Not enough tokens.
  - LoL Dropper (medium):** 10€/2000 seconds of drop. Not enough tokens.
  - LoL Dropper (large):** 25€/10000 seconds of drop. Not enough tokens.
- Others offers:**
  - Blacklist IP:** Options to 'Add' or 'Remove'.

On the right side, there's a sidebar titled 'Account status' with icons for 'Status', 'Recent activity', and 'Log in to support channel'. A green button at the bottom right says 'I have a message'.

## Additional Reading:

J. J. Santanna et al., "Booters — An analysis of DDoS-as-a-service attacks," 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, 2015, pp. 243-251, doi: 10.1109/INM.2015.7140298.

## Slashdot / FlashCrowd effect

- It is not an intended attack (false positives?),
- but effects are the same as for real attacks.
- Popular news spread very quickly
- Social media help information spreading (twitter, facebook, ...)

## Section 4

### Brute-Force

# Brute-Force Attacks / Scans

- Motivation: gain access
- Guessing username or password / scanning hosts
- Dictionary / Enumerated attempts
- Against any protocol or service
- The aim is to gain access (or) steal identity / gather information

## Section 5

### Defense

# Scanning Defense

- Let open only necessary ports
- Drop instead of Reject (politeness vs scanner slowdown?)
- Decoy/Honeypot?

# SYN Flood Defense

- SYN cookies
- allocate system resources only after TCP handshake is completed
- RST cookies
- after received SYN, server sends invalid SYN ACK
- RST should be received from client – in this case the client is valid
- Micro blocks
- only small memory space allocated for incoming SYN requests (e.g. 16 bytes)
- Stack tweaking
- selectively dropping incoming connections
- reducing the timeout when the memory allocated for the connection is freed up

# SYN cookies

- Transmission Control Block (TCB): data structure which holds the connection state information
- Cookie = calculated TCP sequence number

# Smurf attack Defense

- Endpoints should not respond to broadcasted ICMP requests
- Routers should drop the broadcasted ICMP requests

# (D(R))DoS Defense

- Difficult
- Most effective: ISP providing countermeasures
- SYN proxy (until ACK, connection request not forwarded)
- Connection limits (prioritize existing connection, limit per IP etc.)
- Aging (how long can the connection be idle? → TCP RST), timeouts
- Anomaly recognition (malformed headers, protocol state etc.)
- Dark address prevention (address not assigned by IANA most probably spoofed)
- Bandwidth over-subscription
- Blackholing / RTBH
- Load-balancing
- System hardening, tuning (profiling)

# DDoS Defense: Main Issues

- Victim can do completely nothing when the network is under attack.  
**The only hope is contacting ISP (or peering network operator).**
- Victim can easily see the attack X Source network can easily drop the traffic (<http://www.bcp38.info>)
- Lets drop evil packets... **Well, which packets are evil?** This kind of recognition is The Question that can make us rich!
- **Dropping/blocking all means successful attack** — system/service is down or disconnected.
- **World is bigger than we are.**  
There are many devices out there that can be used by attackers, we always have significantly less resources.
- **Spoofed addresses** make harder to find origins of the traffic.

# Brute-Force Attack Defense

- fail2ban
- limited number of login attempts
- good passwords
- key vs password
- unpredictable username (if possible)

## Section 6

Closing Words

# Questions?

## 4. Network Attacks and Their Detection

### Covert Channels, MitM, Poisoning, L7 threats

Simona Buchovecká, Tomáš Čejka

Faculty of Information Technology, CTU in Prague

*simona.buchovecka@fit.cvut.cz, tomas.cejka@fit.cvut.cz*

March 7, 2021

## Section 1

### Covert Channels, Tunneling

## Some definitions of a covert channel:

- a transmission channel that may be used to transfer data in a manner that violates security policy (Van Horenbeeck)
- a means of communication not normally intended to be used for communication (Zander, Armitage & Branch, 2007)
- a mechanism for sending and receiving information data between machines without alerting any firewalls and IDSs on the network (Buetler, 2008)

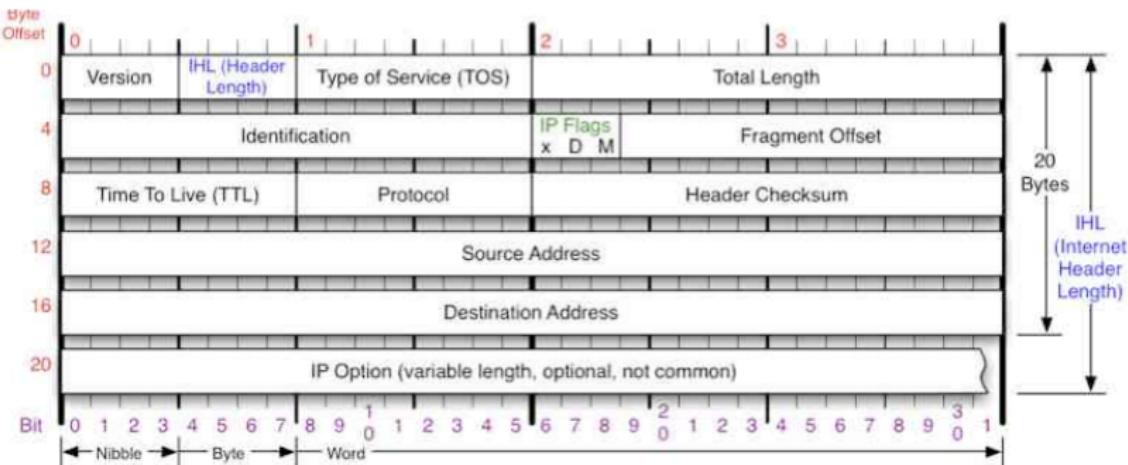
<https://www.sans.org/reading-room/whitepapers/detection/covert-channels-33413>

# Motivation

- Exfiltrate data from an otherwise secure system
- Avoid detection of unauthorized access
- Malware communication — C&C channel hiding
- Circumvent filters which may be in place limiting their freedom of speech
- Bypass firewalls for unrestricted access to the web

# Methods of Covert Data Encoding

- Header bit modulation
- Header bit crafting
- Optional header extension
- Temporal channels



## Section 2

### ICMP

# ICMP Packet

IP Datagram						
	Bits 0–7	Bits 8–15	Bits 16–23	Bits 24–31		
IP Header <b>(20 bytes)</b>	Version/IHL	Type of service	Length			
	Identification		<i>flags and offset</i>			
	Time To Live (TTL)	Protocol	Checksum			
	Source IP address					
	Destination IP address					
ICMP Header <b>(8 bytes)</b>	Type of message	Code	Checksum			
	Header Data					
ICMP Payload <i>(optional)</i>	Payload Data					

# ICMP Tunnelling I

No.	Time	Source	Destination	Protocol	Info	Packet Size
19	11:19:37.039393	10.2.240.197	10.2.240.195	ICMP	Echo (ping) request	770
20	11:19:37.039473	10.2.240.195	10.2.240.197	ICMP	Echo (ping) reply	770
30	11:19:37.071152	10.2.240.197	10.2.240.195	ICMP	Echo (ping) request	770
31	11:19:37.071399	10.2.240.195	10.2.240.197	ICMP	Echo (ping) reply	770
39	11:19:37.652033	[REDACTED].244.0.3	10.2.240.195	DNS	Standard query response CNAME www.l.google.com	192
43	11:19:37.653062	10.2.240.195	[REDACTED].85.227.141	HTTP	GET / HTTP/1.1	738
54	11:19:40.733100	[REDACTED].85.227.147	10.2.240.195	TCP	[TCP segment of a reassembled PDU]	1516
56	11:19:40.733240	[REDACTED].85.227.147	10.2.240.195	TCP	[TCP segment of a reassembled PDU]	1516
58	11:19:40.733870	[REDACTED].85.227.147	10.2.240.195	TCP	[TCP segment of a reassembled PDU]	1516
60	11:19:40.733941	[REDACTED].85.227.147	10.2.240.195	HTTP	HTTP/1.1 200 OK (text/html)	283
66	11:19:40.735795	10.2.240.195	10.2.240.197	ICMP	Echo (ping) reply	1096
67	11:19:40.735906	10.2.240.195	10.2.240.197	ICMP	Echo (ping) reply	1096
68	11:19:40.735983	10.2.240.195	10.2.240.197	ICMP	Echo (ping) reply	1096
69	11:19:40.736059	10.2.240.195	10.2.240.197	ICMP	Echo (ping) reply	1096
70	11:19:40.736134	10.2.240.195	10.2.240.197	ICMP	Echo (ping) reply	1096
71	11:19:40.736209	10.2.240.195	10.2.240.197	ICMP	Echo (ping) reply	1096
72	11:19:40.736306	10.2.240.195	10.2.240.197	ICMP	Echo (ping) reply	1096
73	11:19:40.736381	10.2.240.195	10.2.240.197	ICMP	Echo (ping) reply	1096
74	11:19:40.736479	10.2.240.195	10.2.240.197	ICMP	Echo (ping) reply	1096
75	11:19:40.736552	10.2.240.195	10.2.240.197	ICMP	Echo (ping) reply	1096
76	11:19:40.736633	10.2.240.195	10.2.240.197	ICMP	Echo (ping) reply	1052
77	11:19:40.757953	10.2.240.197	10.2.240.195	ICMP	Echo (ping) request	850
78	11:19:40.758020	10.2.240.195	10.2.240.197	ICMP	Echo (ping) reply	850
92	11:19:40.778930	10.2.240.197	10.2.240.195	ICMP	Echo (ping) request	964
93	11:19:40.778952	10.2.240.195	10.2.240.197	ICMP	Echo (ping) reply	964
94	11:19:40.779003	10.2.240.197	10.2.240.195	ICMP	Echo (ping) request	850
95	11:19:40.779024	10.2.240.195	10.2.240.197	ICMP	Echo (ping) reply	850

# ICMP Tunnelling II

0000	00	00	00	01	00	06	00	50	56	32	74	1c	00	00	08	00	.....P	V2t.....
0010	45	00	02	f2	00	00	40	00	40	01	42	7e	0a	02	f0	c5	E.....8.	B.....
0020	0a	02	f0	c3	08	00	59	f0	e5	ec	00	2e	d5	20	08	80	.....Y.	..... ..
0030	00	00	00	00	00	00	00	00	40	00	00	02	00	00	00	35	.....	@.....5
0040	00	00	02	b9	00	2e	e5	ec	47	45	54	20	68	74	74	70	.....	GET http
0050	3a	2f	2f	77	77	77	2e	67	6f	6f	67	6c	65	2e	63	6f	://www.g	oogle.co
0060	6d	2f	20	48	54	54	50	2f	31	2e	31	0d	0a	48	6f	73	m/	HTTP/ 1.1..Hos
0070	74	3a	20	77	77	77	2e	67	6f	6f	67	6c	65	2e	63	6f	t:	www.g oogle.co
0080	6d	0d	0a	55	73	65	72	2d	41	67	65	6e	74	3a	20	4d	..User-	Agent: M
0090	6f	7a	69	6c	6c	61	2f	35	2e	30	20	28	58	31	31	3b	ozilla/5 .0 (X11;	
00a0	20	55	3b	20	4c	69	6e	75	78	20	69	36	38	36	3b	20	U; Linu	x i686;
00b0	65	6e	2d	55	53	3b	20	72	76	3a	31	2e	39	2e	31	2e	en-US; r	v:1.9.1.
00c0	38	29	20	47	65	63	6b	6f	2f	32	30	31	30	30	32	31	8)	Gecko /2010021
00d0	34	20	4c	69	69	75	78	20	4d	69	6e	74	2f	38	20	28	4	Linux Mint/8 (
00e0	48	65	6c	65	6e	61	29	20	46	69	72	65	66	6f	78	2f	Helena)	Firefox/ 3.5.8..A
00f0	33	2e	35	2e	38	0d	0a	41	63	63	65	70	74	3a	20	74	ccept: t	ext/html , applica
0100	65	78	74	2f	68	74	6d	6c	2c	61	70	70	6c	69	63	61	=0.9,*/* ;q=0.8..	=0.9,*/* ;q=0.8..
0110	74	69	6f	6e	2f	78	68	74	6d	6c	2b	78	6d	6c	2c	61	Accept-L	anguage: en-us,e n;q=0.5.
0120	70	70	6c	69	63	61	74	69	6f	6e	2f	78	6d	6c	3b	71	.Accept-	Encoding: gzip,deflate..
0130	3d	30	2e	39	2c	2a	2f	2a	3b	71	3d	30	2e	38	0d	0a	Accept-C	harset: ISO-8859 -1,utf-8
0140	41	63	63	65	70	74	2d	4c	61	6e	67	75	61	67	65	3a	;q=0.7,*	;q=0.7..
0150	20	65	6e	2d	75	73	2c	65	6e	3b	71	3d	30	2e	35	0d	Koen-Ali	ua: 300.
0160	0a	41	63	63	65	70	74	2d	45	6e	63	6f	64	69	6e	67		
0170	3a	20	67	7a	69	70	2c	64	65	66	6c	61	74	65	0d	0a		
0180	41	63	63	65	70	74	2d	43	68	61	72	73	65	74	3a	20		
0190	49	53	4f	2d	38	38	35	39	2d	31	2c	75	74	66	2d	38		
01a0	3b	71	3d	30	2e	37	2c	2a	3b	71	3d	30	2e	37	0d	0a		
01b0	4b	55	45	70	2d	41	6c	60	76	65	7c	70	72	70	70	6f		

## Section 3

### Domain Name System (DNS)

## Recap: How does DNS work?

```
1252:~# dig fit.cvut.cz

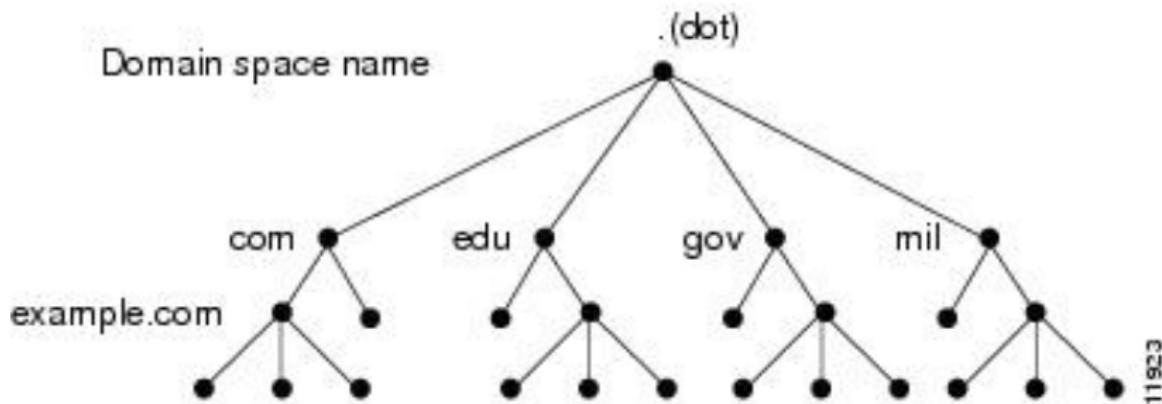
; <>> DiG 9.11.4-P1-RedHat-9.11.4-5.P1.fc28 <>> fit.cvut.cz
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61156
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
;; ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 8192
;; QUESTION SECTION:
;fit.cvut.cz.      IN  A

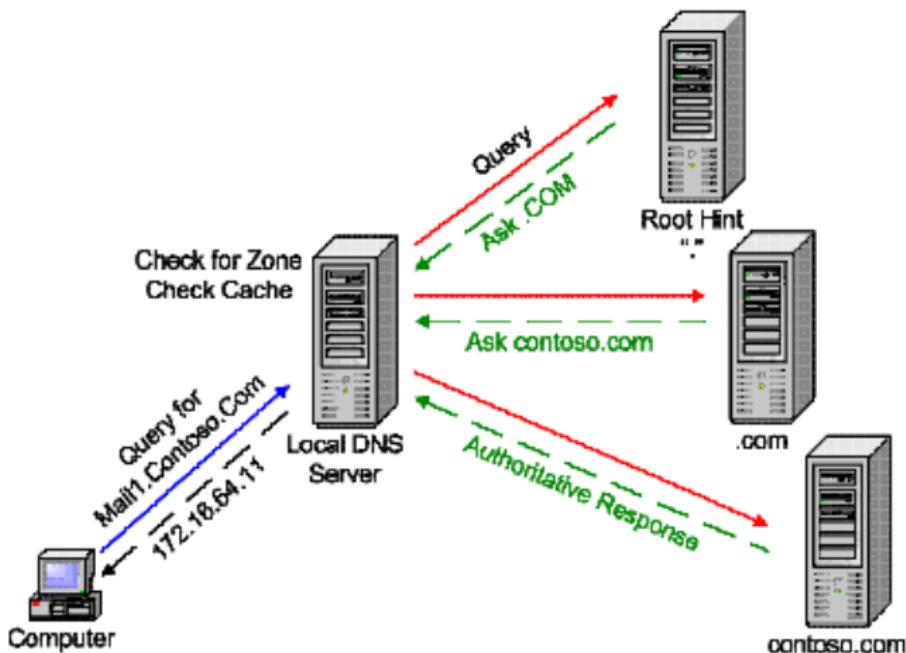
;; ANSWER SECTION:
fit.cvut.cz.    3327   IN  A 147.32.232.248

;; Query time: 1 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Sun Oct 21 12:52:22 CEST 2018
;; MSG SIZE  rcvd: 56
```

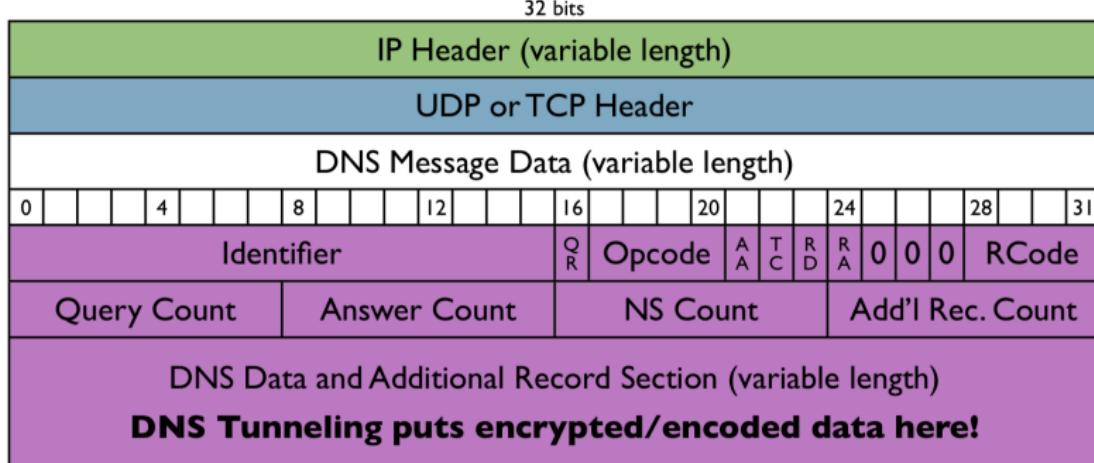
# DNS I



# DNS II



# DNS Message



QR ... Query / Response

RD ... Recursion Desired

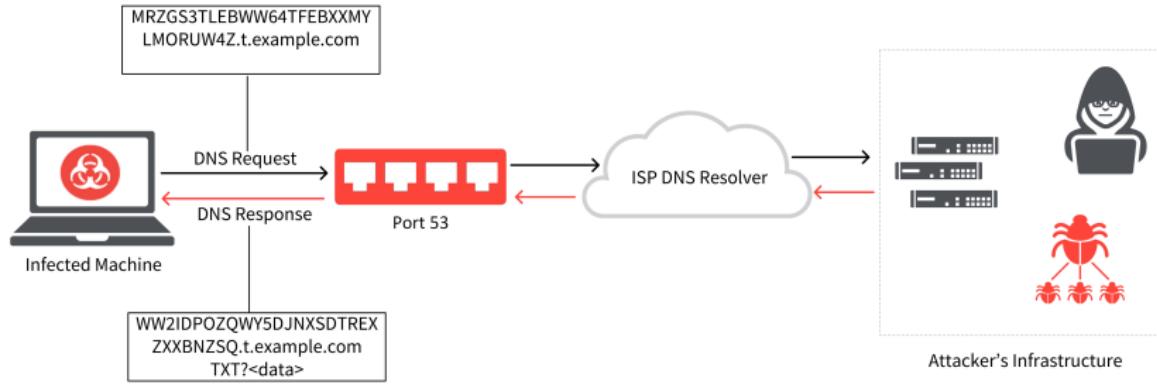
AA ... Authoritative Response

RA ... Recursion Available

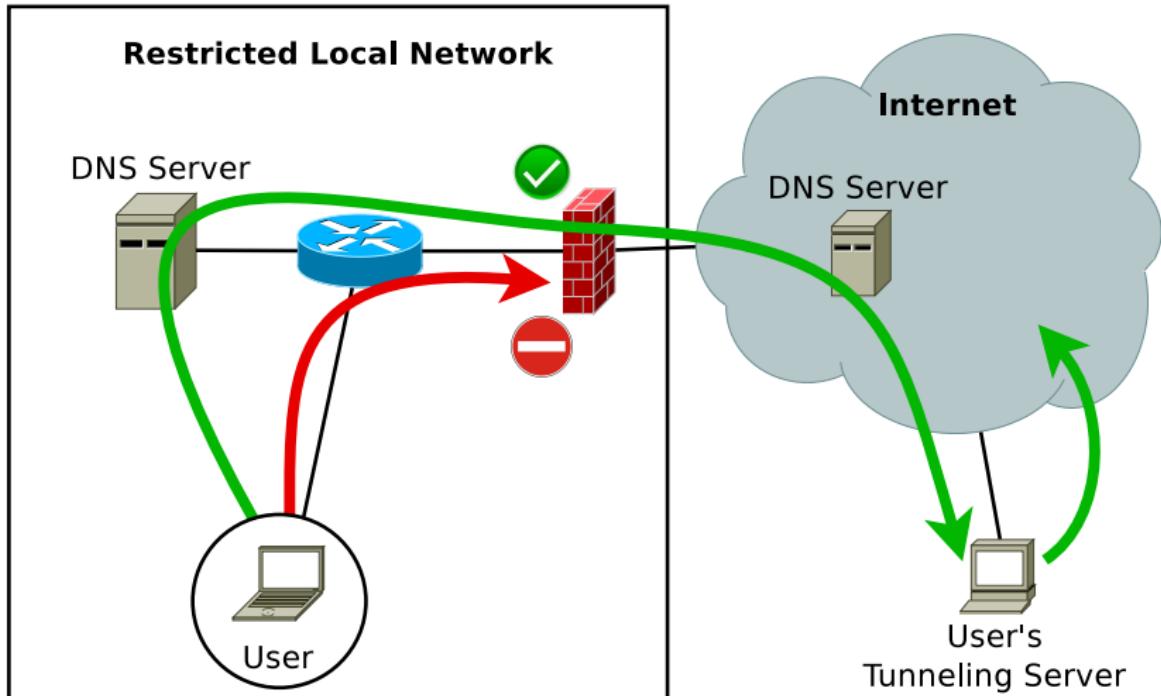
TC ... Truncation Flag (UDP<512B)

RCode ... Response Code (0-10)

# DNS Tunnel (malware communication)



# DNS Tunnel (escape from restricted network)



# DNS Tunnelling: HowTo

## Transmit “Greetings!” ...

- Client: base32 encode "Greetings!" and make a DNS query:  
I5ZGKZLUNFXGO4ZB.sshdns.mydomain.com
- Local DNS response: "I do not know I5ZGKZLUNFXGO4ZB.  
sshdns.mydomain.com, ask the DNS server of mydomain.com at IP  
address: w.x.y.z."
- w.x.y.z DNS response: "Ask DNS for sshdns.example.com at IP:  
a.b.c.d". This is attackers server with the proxy software.
- The proxy server base32 decodes I5ZGKZLUNFXGO4ZB as  
"Greetings!"

## ... and receive a reply “Hello...”

- The proxy server base64 encodes "Hello..." to get "SGVsbG8uLi4=" and returns that in a TXT record.
- The client receives "SGVsbG8uLi4=" and b64decode's it to get "Hello..."

# Example of Tunneled Data I

## DNS Request

Daaapiaicab.FV++++++9-J8C8FR3bL+P3L+ZLPb2XZCvg7LYN  
qwo-BvjMj0Dlt4U91.sv7KFx672PumRw8Zkz2gZWUaFhuNaK0fQ2  
IsVKRZMh5I3vp5U1aq05qQV.o8ht+jU2qSNm5rqNbdXdDPTnaf8a  
391UYGOfFV2JE8l06JaJ0XDdDoSkg.DAC-GMaj7klra4TVy3+bnT  
09j14lhIk+AkavZiqgKy3fjakMjSzIDgKvg.abc.ab

= 255 characters of domain name

## Example of Tunneled Data II

### DNS Request

Paaapiamci1gq.abc.ab

Paaapiamei1ia.abc.ab

Paaapiaici1lq.abc.ab

Paaapiaiei1mq.abc.ab

= many packets with short domain names lookups

### DNS Response

fp4suaacaakjngaaaеaaaagsaqaabhh6ovo2cp3kedmpt7ieaeaa

aaa3aaaabsm.wvxuacaaiaaaqbuikomje3t7uab3vmf55ydjxg47

jlyphl7y.v3.url.abc.ab

data stored in TXT field

# Detection

- Unencrypted tunnel:  
Search for specific signatures, e.g., SSH connection contains identification string  
(SSH-protoversion-softwareversion SP comments CR LF).
- Anomaly detection
  - abnormally big packets — ICMP, DNS
  - high packet rate of ICMP, DNS
- Entropy of DNS data?

# Detection

T. Cejka, Z. Rosa, and H. Kubatova: *Stream-wise detection of surreptitious traffic over DNS*. In 2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Athens, Greece, 2014, pp. 300–304.

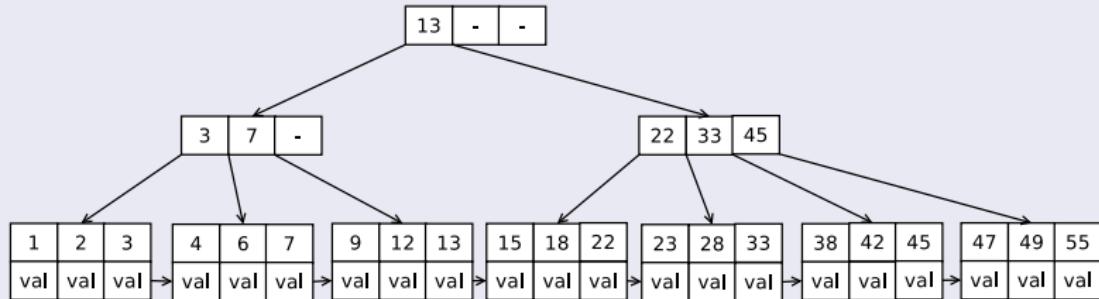
## What the Detection Module Analyzes?

Many characteristics are observed or computed:

- ① Mean value and variance of sizes of DNS requests & responses
- ② Number of letters/digits in domain names
- ③ Fraction of common part of domain names
- ④ Repeating domain names

# Data Structures

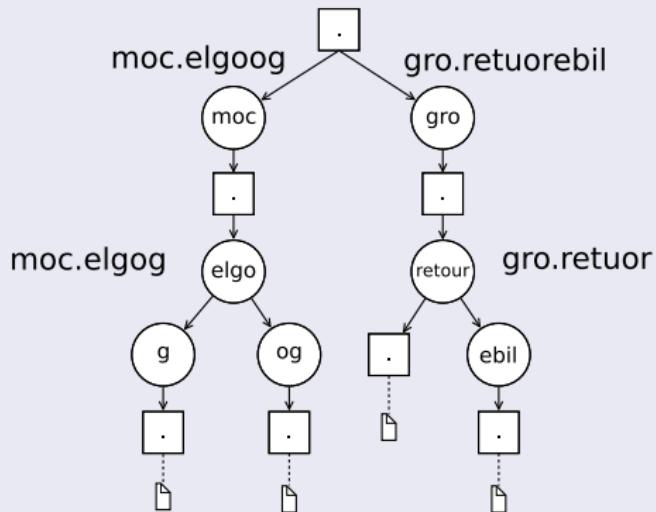
## B+ tree



- network (IPv4 / IPv6) addresses used as keys
- stored values contain information about DNS traffic of an address

# Data Structure in the memory

## “Prefix” tree



- used for domain names storage (from right to left)
- analysis of different and common parts of domain names
- extended with metadata (statistics about domain names)

# Countermeasures?

- Rate limiting
- Implicit output blocking & Permitted local proxy (with analysis)
- ???

## Section 4

DNS over HTTPS

# DNS over HTTPS (DoH)

- Encrypted communication of DNS requests&responses using HTTPS
- Motivation: “privacy” of the users
- Based on GET / POST methods
- wireformat — binary DNS data encoded to HTTPS data
- DoH providers, e.g., Cloudflare, Google
- Supported by modern web browsers, OS

# DoH: Potential Security Threats

- Covert channel
- CC communication
- Serving malware files
- Serving hidden links

# DoH: Research in Detection

- DoH traffic differs in some characteristics
- Firefox / Chrome / cloudflared differs
- Beware of traffic context

D. Vekshin, K. Hynek, and T. Cejka: *DoH Insight: Detecting DNS over HTTPS by Machine Learning*. In Proceedings of the 15th International Conference on Availability, Reliability and Security, New York, NY, USA, 2020.

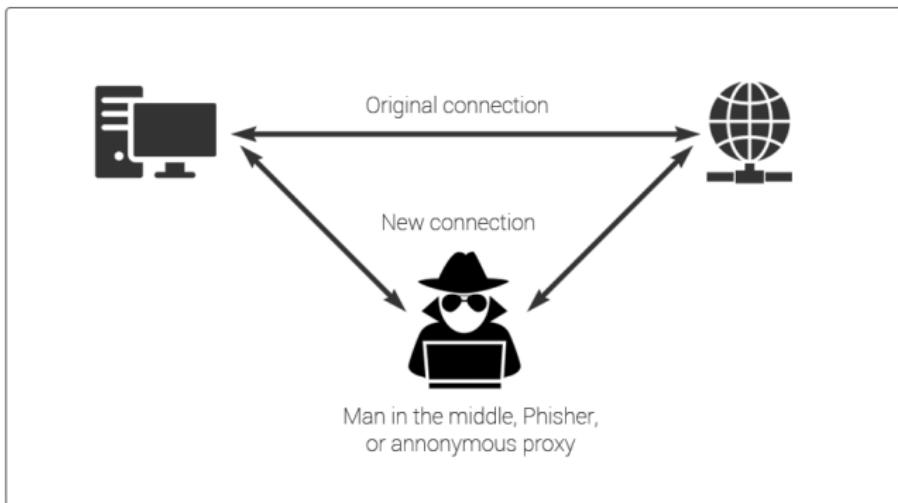
K. Hynek and T. Čejka: *Privacy Illusion: Beware of Unpadded DoH*. Proceedings of the 11th Annual IEEE Information Technology, Electronics and Mobile Communication Conference (IEMCON2020), 2020.

## Section 5

### Man in the Middle (MitM)

# MitM Introduction

- Goal: to allow the intruder or the unauthorized party to eavesdrop and/or modify the transmitted data
- 1<sup>st</sup> step: getting the access to the network traffic



- Mr.Robot series: “femtocell”: bogus AP
- DHCP – rogue DHCP server
- ARP cache poisoning
- DNS cache poisoning
- DNS hijacking ([https://en.wikipedia.org/wiki/Domain\\_hijacking](https://en.wikipedia.org/wiki/Domain_hijacking))
- BGP hijacking ([https://en.wikipedia.org/wiki/BGP\\_hijacking](https://en.wikipedia.org/wiki/BGP_hijacking))
- Session hijacking ([https://en.wikipedia.org/wiki/Session\\_hijacking](https://en.wikipedia.org/wiki/Session_hijacking))
- NTP MitM (Delorean <https://github.com/PentesterES/Delorean>), affecting HSTS?

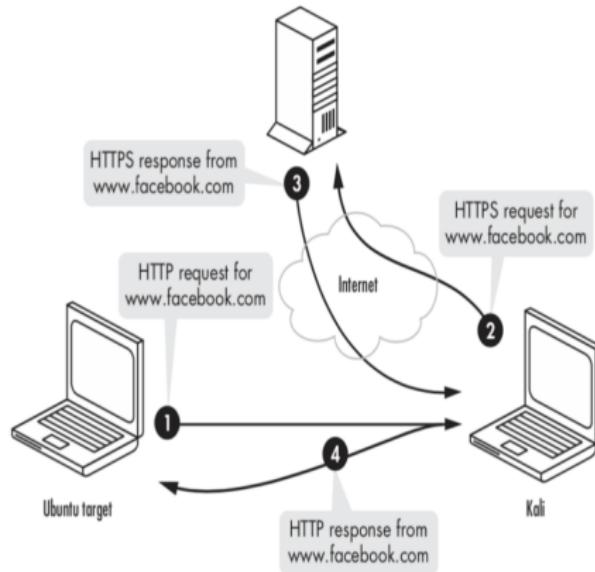
Selvi, Jose. Bypassing HTTP strict transport security. Black Hat Europe (2014).

## How to intercept the encrypted data?

- SSL Stripping
- Host Certificate Hijacking / Certificate pinning
- TLS Protocol Downgrade

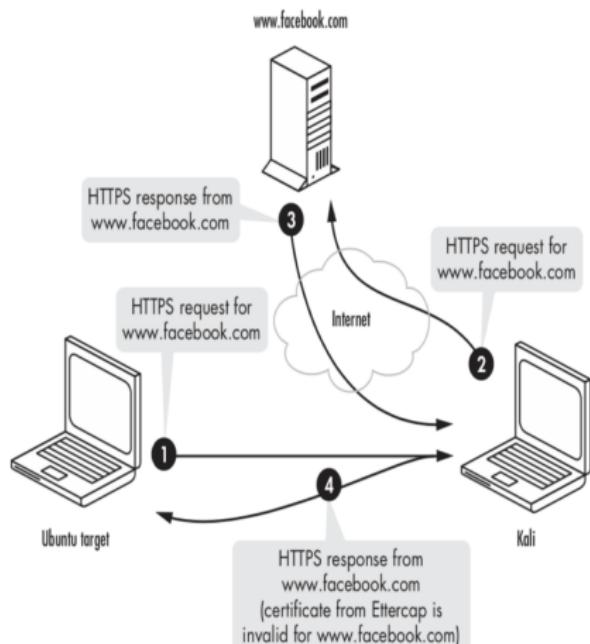
# SSL Stripping

- SSLstrip tool
- In SSL MitM users have to click on SSL certificate warning
- Connection to the user is SSL stripped in plain HTTP
- Connection to the webserver from the attacker via HTTPS



# Host Certificate Hijacking / Certificate Pinning

If the attacker is able to inject malicious root certificate into the trusted root certificate authority store of the victim device, user will receive no warning messages for certificates being not valid.



# TLS Protocol Downgrade

Manipulate the negotiated connection to downgrade the negotiated protocol or cipher suites — various known attacks

- BEAST (CVE-2011-3389)
- BREACH (CVE-2013-3587)
- CRIME (CVE-2012-4929)
- Heartbleed (CVE-2014-0160)
- POODLE (CVE-2014-3566)

## Section 6

### DNS Attacks

- To increase performance – server caches resolved translation for a certain amount of time
- No cryptographic protection, no authentication, no integrity checks by default
- DNS accepts only responses to pending queries... However, first good answer wins
- Cache poisoning attacks possible

Sooel Son and Vitaly Shmatikov: *The Hitchhiker's Guide to DNS Cache Poisoning*

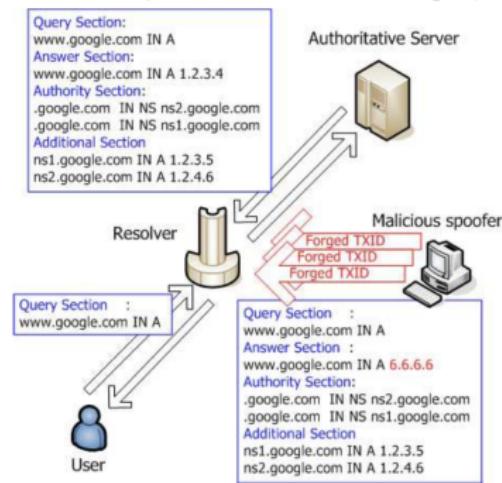
# Blind response forgery using birthday attack

By default, DNS only checks are:

- 16-bit transaction ID (TXID) must match: query and response
- *srcip* and *dstport* of response ? = *dstip* and *srcport* of the query
- First arriving UDP packet satisfying the condition is treated as valid

TXID has only N bits of entropy (theoretically N=16);

Much less in practice: TXID not randomized properly (just incrementing)



# Cache Poisoning without Response Forgery

## Bailiwick rule

(example from <http://www.linuxjournal.com/content/understanding-kaminsky-s-dns-bug>)

- Adopted by BIND in 1993
- Before adopted, the owner of any DNS authoritative server could compromise records corresponding to any domain name
- Kaminsky's exploit presented on BlackHat conference allows to even bypass the bailiwick check

```
# dig doesnotexist.example.com
;; ANSWER SECTION:
doesnotexist.example.com. 120 IN A 10.10.10.10

;; AUTHORITY SECTION:
example.com. 86400 IN NS www.example.com.

;; ADDITIONAL SECTION:
www.example.com. 604800 IN A 10.10.10.20
```

# Fragmentation Attack

https:

//www.incapsula.com/ddos/attack-glossary/ip-fragmentation-attack-teardrop.html

https://ripe67.ripe.net/presentations/240-ipfragattack.pdf

## IP Fragmentation and Reassembly (Example)



**Length** - The size of the fragmented datagram

**ID** - The ID of the datagram being fragmented

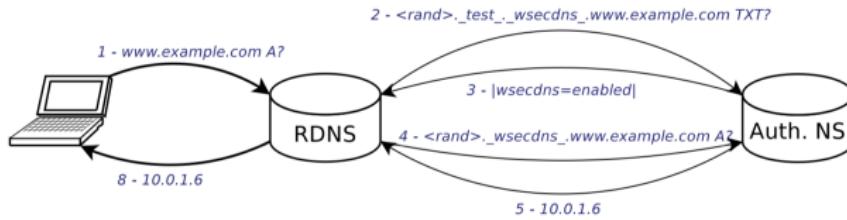
**Fragflag** - Indicates whether there are more incoming fragments

**Offset** - Details the order the fragments should be placed in during reassembly

# Response forgery using eavesdropping

Different proposed countermeasures:

- Source port randomization
- 0x20-bit encoding — randomize the case of question name — ask for wWw.xyz.com, WWW.XyZ.com instead of www.xyz.com etc.
- Extended Query ID (XQID) - A domain name label consisting of 24 to 63 random characters (0-9, a-z). Not case sensitive
- WSEC-DNS (<http://ieeexplore.ieee.org/document/5270363/>):
  - Does not add confidentiality nor provides authentication
  - Protects against “blind” brute forcing
  - DNS is still vulnerable to trivial attacks such as network eavesdroppers & packet forging



## Section 7

Brief information about DNSSEC

# DNSSEC

[https://www.nanog.org/meetings/nanog51/presentations/Sunday/  
DNSSEC-tutorial-for-NANOG51-2011-01.pdf](https://www.nanog.org/meetings/nanog51/presentations/Sunday/DNSSEC-tutorial-for-NANOG51-2011-01.pdf)

- DNSSEC provides no confidentiality = all DNS data are public
- DNSSEC provides data origin authentication & data integrity
- Each zone has a public/private key pair
- Public key is stored in DNSKEY record
- Private key needs to be kept safe — HSM (Hardware Security Module)

# DNSKEY record

256/257 — 16bit flag field — DNSSEC zone key/key-signing key (used for signing zone keys) 3 protocol octet for DNSSEC 5 algorithm number (RSA with SHA-1) The public key itself

```
example.com. 2849 IN DNSKEY 257 3 8 (
    AwEAAZ0aqu1rJ6orJynrRfNpPmayJZoAx9Ic2/R19VQW
    LMHyjxxem3VUSoNUIFXERQbj0A90gp0zDM9YIccKLRd6
    LmWiDCt7UJQxVdD+heb5Ec4qlqGmyX9MDabkvX2NvMws
    UecbYBq8oXeTT9LRmCUt9KUT/W0i6DKECxoG/bWTykrX
    yBR8e1D+SQY430AVjlWrVltHxgp4/rhBCvRbmdflunaP
    Igu27eE2U4myDSLTA8a4AOrB5uHG4Pk0a9dIRs9y00M2m
    Wf4lyPee7vi5few2dbayHXmieGcaAHrx76NGAABeY393
    xjlmDNCUkF1gpNWUla4fWZbbaYQzA93mLdrng+M=
)
; KSK; alg = RSASHA256 ; key id = 45620
example.com. 2849 IN DNSKEY 256 3 8 (
    AwEAAAd3ls8XH4tS6n576cFPy9ZbtQlf8ivP29WA41Kes
    7KRQvU+jAT1R68mBW2AaIMxfdYaV9ddg0zz6jAt8o3zT
    foylcr8UpmgDOC1qZ/0QYQ/gAOATMDCT6lz8cz+eYB+R
    k2b/Ptuhkx2HRkZJKJyirRyHg7vYQ0gMIdNJ8D9munn
)
; ZSK; alg = RSASHA256 ; key id = 63855
```

# RRSIG Record

- Each resource record set (RRSET) in a zone is signed by zone's private key
- RRSET – records with same owner, class and type
- Digital signature is stored in RRSIG record

A type of records signed 5 digital algorithm used (RSA with SHA1) 3 number of labels in the signed name 86400 original TTL When the signature expires When the record were signed 41148 the key ID/tag/footprint Test.com signer's name Digital signature itself

```
; ; ANSWER SECTION:  
example.com.      16484 IN A 93.184.216.34  
example.com.      16484 IN RRSIG A 8 2 86400 ( 20181029053351 20181007181129 63855 example.com.  
XRKd78dE8RDam/6g2gSRM3GRy8PvpAoNFMZRPJSSMjRn  
lftP9aNZHGuKldks/2R6f7hY/iZpzuwsI3LEv8T7e1DV  
T1QesVe5S0StdkD+ssFq9iG+Rdbe7cjQDQ0aWAKzgAPE  
bFmh/2wZ4NT1CQKoshprVqcpMeasCx1JvXCyB7c= )
```

# Issues with DNSSEC

- Large response (packet-amplifier)
- Complexity of key management – prevents broad deployment
- Subdomain Injection
  - causes resolvers to accept, cache and provide to clients a mapping for a non-existing (child) domain, of a DNSSEC-protected parent domain
- Attacker can create fake sub-domains – this can lead to XSS, phishing, cookie stealing
- Name Server Hijacking
  - causes resolvers to cache and use incorrect name servers for a DNSSEC-protected domain point them to name servers belonging to the attacker

## Section 8

### Some Other L7 Threats

- Deregistering
- INVITE of Death (malformed or otherwise malicious SIP INVITE,  
[https://en.wikipedia.org/wiki/INVITE\\_of\\_Death](https://en.wikipedia.org/wiki/INVITE_of_Death))
- Password guessing
- User account (extension) scanning
- Dial scheme guessing (exploit of weak configuration)
- SCAM (similar to SPAM known from e-mails) → leads to DoS

## Additional Reading:

- T. Jánský, et al.: *Hunting SIP Authentication Attacks Efficiently*. AIMS 2017, Zurich.
- T. Cejka, et al.: *Using Application-Aware Flow Monitoring for SIP Fraud Detection*.  
AIMS 2015, Ghent.

# Shellshock / Bashdoor

- Vulnerability in Bash, disclosed in 2014
- Arbitrary code execution
- Specific exploitation vectors:
  - CGI-based web server
  - OpenSSH server, *ForceCommand* feature
  - DHCP clients, can pass commands to Bash, additional options
  - Qmail server, processing mails by Bash
  - IBM HMC, gain access to Bash from the restricted shell of the IBM Hardware Management Console

# HTTP

- SlowLoris ([https://en.wikipedia.org/wiki/Slowloris\\_\(computer\\_security\)](https://en.wikipedia.org/wiki/Slowloris_(computer_security))) / Slowdroid (<https://en.wikipedia.org/wiki/Slowdroid>)
- RUDY (R-U-Dead-Yet? <https://sourceforge.net/projects/r-u-dead-yet/>)
- Exploit known vulnerabilities (CMS, ...)
- Password guessing (there was a “bug” in WordPress: different delays during authentication)
- SQL injection over HTTP(s)

# SMTP

- Sending SPAM is one of the typical activities of malware.
- Spoofed addresses make a SMTP server to produce lots of bounces.
- Spreading of malware, worms
- There are many blacklists (<http://valli.org>) and best practices as a “defense”.

## Section 9

Closing Words

## References

- [http://cs.uccs.edu/~jkalita/papers/2014/  
HoqueNetworkAttacksJCNA2014.pdf](http://cs.uccs.edu/~jkalita/papers/2014/HoqueNetworkAttacksJCNA2014.pdf)
- <https://www.eecis.udel.edu/~sunshine/publications/CCR.pdf>
- [https://www.eecis.udel.edu/~mills/teaching/eleg867b/  
dos/p38-paxson.pdf](https://www.eecis.udel.edu/~mills/teaching/eleg867b/dos/p38-paxson.pdf)
- [http://www.cs.uccs.edu/~jkalita/papers/2013/  
BhuyanMonowarComputerJournal2013.pdf](http://www.cs.uccs.edu/~jkalita/papers/2013/BhuyanMonowarComputerJournal2013.pdf)
- [http://www.cs.uccs.edu/~jkalita/papers/2015/  
RupDekaJNCA2015.pdf](http://www.cs.uccs.edu/~jkalita/papers/2015/RupDekaJNCA2015.pdf)
- [http://ce.sharif.edu/courses/83-84/1/ce534/resources/  
root/Papers/attackstaxonomy.pdf](http://ce.sharif.edu/courses/83-84/1/ce534/resources/root/Papers/attackstaxonomy.pdf)
- [https://en.wikipedia.org/wiki/Timeline\\_of\\_computer\\_  
security\\_hacker\\_history](https://en.wikipedia.org/wiki/Timeline_of_computer_security_hacker_history)
- and other links inside this presentation

# Questions?

## 5. Secure Remote Access

Simona Buchovecká, Tomáš Čejka

Faculty of Information Technology, CTU in Prague

*simona.buchovecka@fit.cvut.cz, tomas.cejka@fit.cvut.cz*

November 1, 2020

# Securing Remote Access Virtual Private Networks

## Disscussion

Private vs. Non-Private Networks

**Offices in several cities:**

- Internet – problems with confidentiality, integrity, authentication
- MPLS – how to connect home/mobile users?

# Virtual Private Networks #1

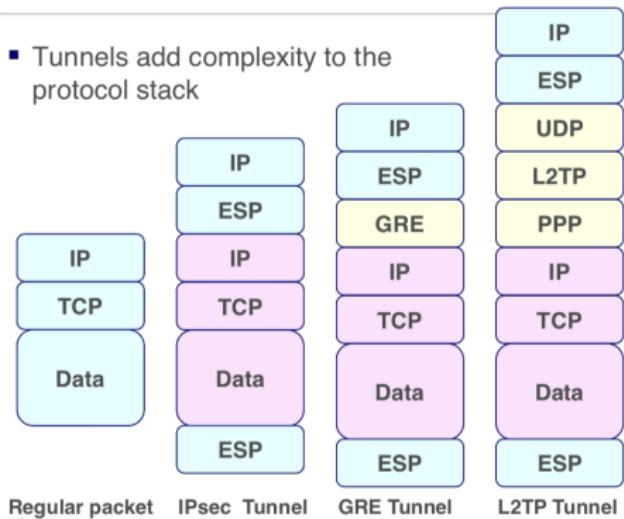
- ① Even with the ability to sniff my packets on the Internet, you can't tell what I'm sending.
- ② They can't modify my packets without me knowing it.
- ③ They can't send me traffic and have me think it came from one of my sites.
- ④ They can drop some of my packets, but they can't drop a class of packets (Because of #1)

# Virtual Private Networks #2

## Basic Idea:

- **Tunnelling & Encryption**
- GRE/L2TP
- IPsec/IKE
- SSL

- Tunnels add complexity to the protocol stack



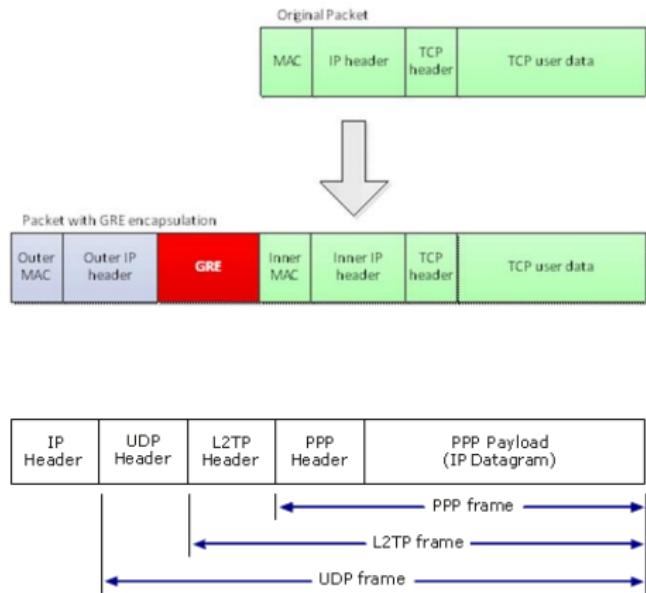
# GRE/L2TP

**GRE – Generic Routing and Encapsulation adds a simple header to traffic, and has its own protocol number (47)**

- Used mainly by Cisco for site-2-site VPNs.

**L2TP – Layer-2 Transport Protocol establishes a point-to-point protocol (PPP) link over UDP port 1701.**

- Used mainly for remote access in Windows, Mac, iOS, Android



# IPSec #1

- To establish a secure IPSEC connection two nodes must execute a key agreement protocol.
  - The sub-protocol of IPSEC that handles key negotiations is called IKE (Internet Key Exchange).
- First, assume two nodes have agreed on keys (via IKE) and see how they proceed to protect their communication via IPSEC
- An IPsec protected connection is called a security association.
  - IPsec is a level-3 protocol (runs on top of IP), and below TCP/UDP
  - Security associations may either be end-to-end or link-to-link.
- Two modes of encapsulating IPsec data into an IP packet define two modes of operation:
  - Transport mode and tunnel mode

## Authentication Header (AH)

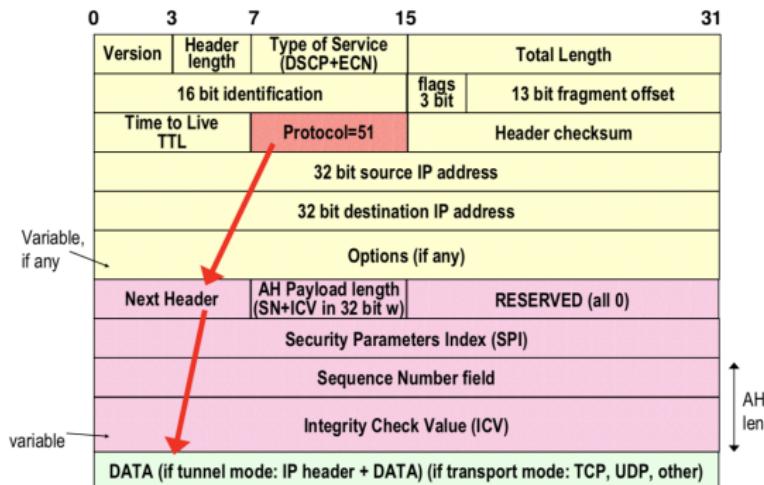
- Provides packet authentication using a keyed MAC function
- Ensures that the packet actually came from the peer with which you exchanged keys, was not modified, and was not replayed.

## Encapsulated Security Protocol (ESP)

- Provides packet authentication and encryption.
- Uses a keyed MAC and an encryption function.
- Ensures your traffic cannot be read en route.

# Authentication Header

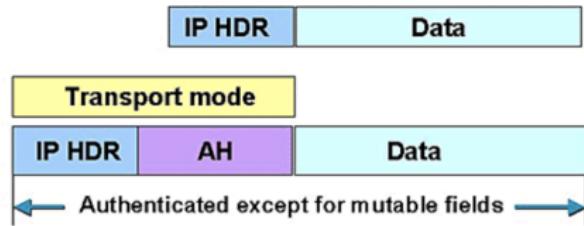
**Sequence number**  
(no seq. no. in IP header)  
prevents replays of  
authenticated packets



# Transport Mode — Layers 4–7

## Transport mode:

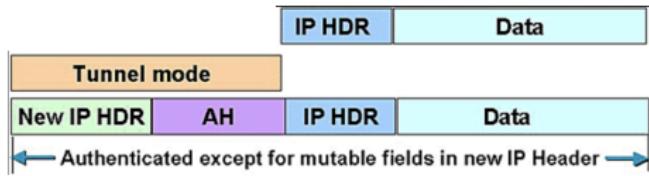
- was designed to save bandwidth in end-to-end associations
- payload is typically encrypted and authenticated
- IP header is unencrypted, and may or may not be authenticated



# Tunnel Mode — Layers 3–7

## Tunnel mode:

- protects both the payload and IP header of the original packet
- if encryption is used between gateways in tunnel mode, then it reduces information for traffic analysis



# SSL VPNs

- In its simplest form, an SSL VPN is a portal that gives the user access to company resources through a web interface:
  - Email
  - Intranet
  - Fileshares, anything
- Advantages over IPsec tunnels
  - Clientless – all you need is a browser
  - Web-based authentication
  - It's HTTPS and most firewalls allow HTTPS

- **Clientless** — Clientless mode provides secure access to private web resources and will provide access to web content. This mode is useful for accessing most content that you would expect to access in a web browser, such as Internet access, databases, and online tools that employ a web interface.
- **Thin client (port-forwarding Java applet)** — Thin-client mode extends the capability of the cryptographic functions of the web browser to enable remote access to TCP-based applications such as Post Office Protocol version 3 (POP3), Simple Mail Transfer Protocol (SMTP), Internet Message Access protocol (IMAP), Telnet, and SSH.
- **Tunnel mode** — Full-tunnel client mode offers extensive application support through its dynamically downloaded Cisco AnyConnect VPN Client (next-generation SSL VPN Client) for SSL VPN. Full tunnel client mode delivers a lightweight, centrally configured and easy-to-support SSL VPN tunneling client that provides network layer access to virtually any application.

## OpenVPN

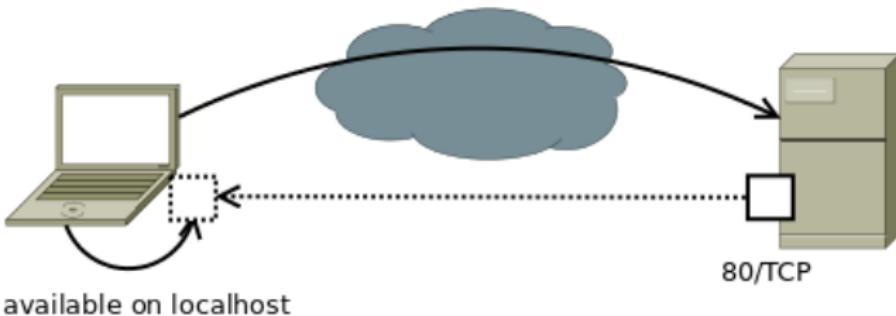
- Based on OpenSSL
- Uses UDP or TCP
- TUN (encapsulates from L3) or TAP (encapsulates from L2)
- various types of authentication (no, shared key, certificates)
- Topology /30 connections, P2P, subnet

# Tunneling

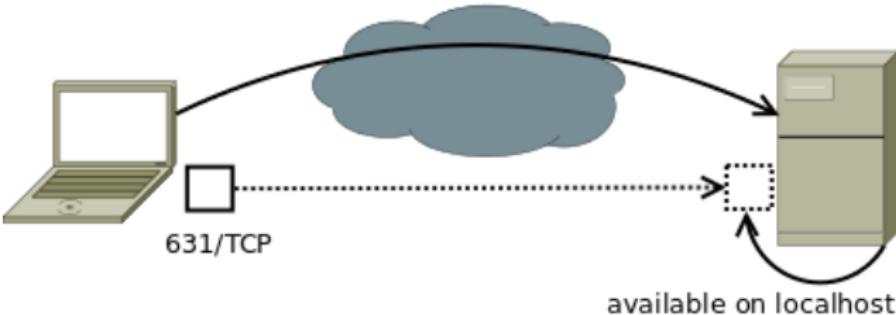
- stunnel (<https://en.wikipedia.org/wiki/Stunnel>)
- SOCKS (<https://en.wikipedia.org/wiki/SOCKS>)
- socat (<http://www.dest-unreach.org/socat/>)  
socat - openssl-listen:12346,method=TLS1.2,key=server.key,  
cert=server.crt,cafile=ca.crt,reuseaddr,fork  
socat udp4-listen:4739,reuseaddr,fork openssl:localhost:12346,  
key=client.key,cert=client.crt,cafile=ca.crt,commonname=nemea-server
- SSH
  - -L (open local port),
  - -R (open remote port),
  - -D (open local SOCKS proxy)

# SSH Port Forwarding

SSH Connection -L 1234:localhost:80 server

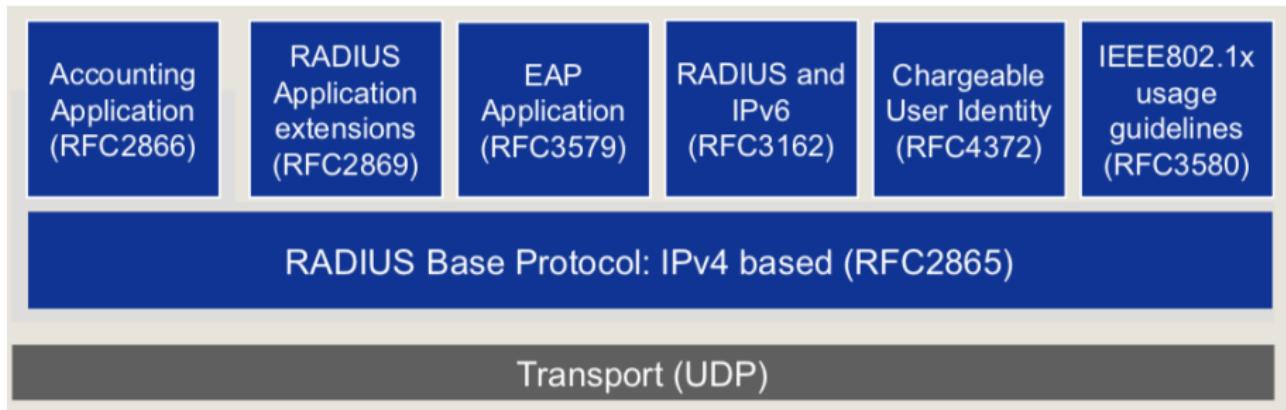


SSH Connection -R 1234:localhost:631 server



# RADIUS

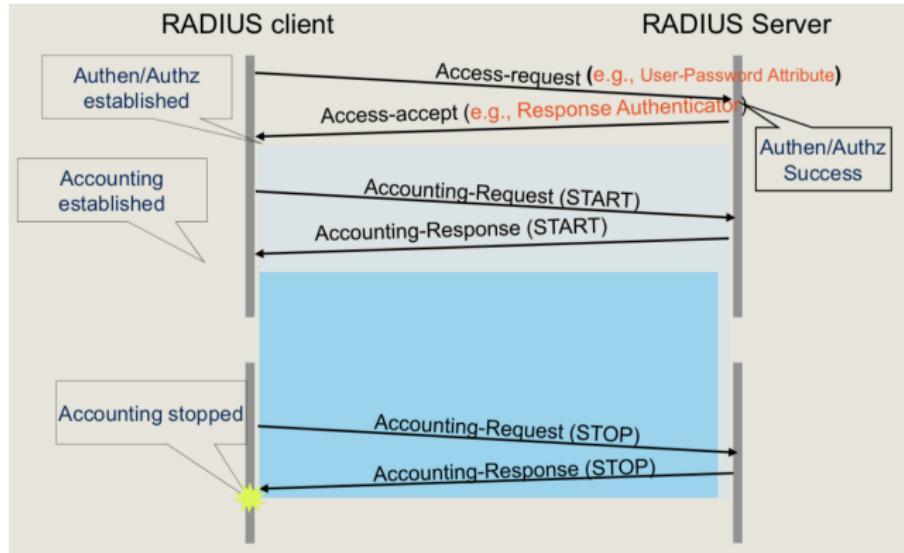
- Motivation: single user database
- Provides centralized AAA functionalities
  - Authentication
  - Authorization
  - Accounting
- Based on UDP/IP – server port 1812, client port ephemeral



# RADIUS - Basic Operations

## Message types:

- 1 Access-Request
- 2 Access-Accept
- 3 Access-Reject
- 4 Accounting-Request
- 5 Accounting-Response
- 11 Access-Challenge
- 12 Status-Server (experimental)
- 13 Status-Client (experimental)
- 255 Reserved



# Architecture

- RADIUS server
- Users DB
  - Authentication information + authentication method
- Clients DB
  - Clients allowed to communicate with the server
- Accounting DB
- Per-packet authenticated reply (using shared key)
- Encrypted user password transmission (same shared key), other info transmitted in plaintext

# RADIUS and UDP

## Advantages

- If request to primary authentication server fails, secondary server must be queried:
  - copy of request must be kept above transport layer
  - retransmission timers still required, above transport layer
- Stateless nature of RADIUS protocol within communication network simplifies use of UDP:
  - transport connection between client/server remains even if network failures are occurring

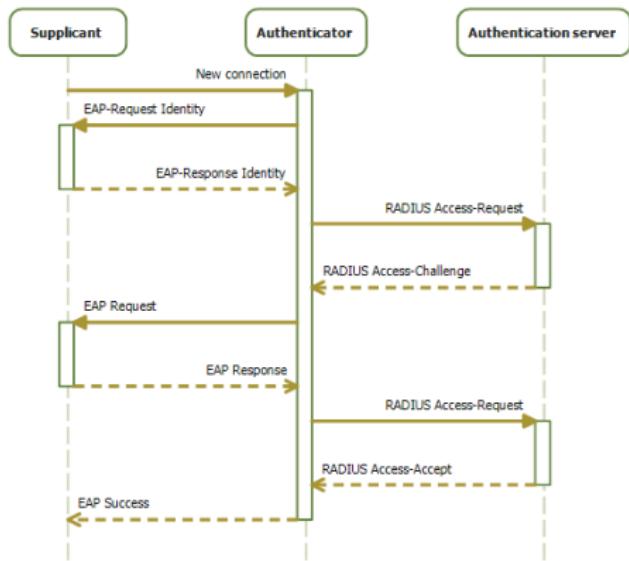
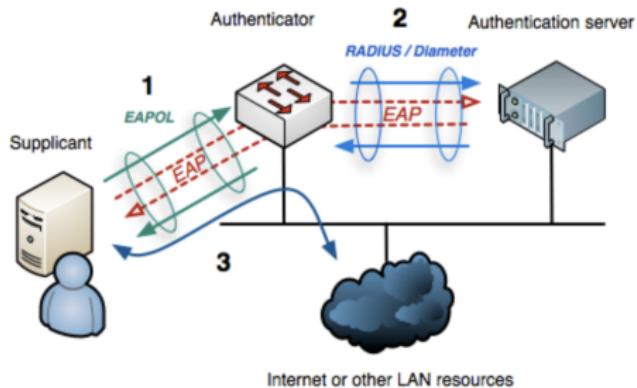
## Disadvantages

- Transport is not reliable (layer above transport has to take care of packet losses)
- TCP adapts to network congestion, while UDP does not

# RADIUS Weaknesses

- Vulnerable to message sniffing and modification
- Clear-text protocol - privacy issues
- Access-Request not authenticated
  - MITM possible, access request forgery possible
- Various attacks presented
  - Shared secret attacks
  - Offline dictionary attacks
  - PRNG attacks

# RADIUS - Use in 802.1x Authentication



# References

- William Stallings: Data and Computer Communications
- Yoav Nir: Lecture on PKI, SSL and VPN (Information Security – Theory vs. Reality)
- Aiko Pras et. Al: Lecture on AAA protocols
- Giuseppe Bianchi: Lectures on Handling Remote Access: RADIUS; RADIUS limits and extensions
- Giuseppe Bianchi: Lecture on IPSec Basics
- Phil Scott: Lecture on Virtual Private Networks
- Wikipedia: RADIUS, 802.1x

# Questions?

## 9. Active Defense, Cyber Deception

Simona Buchovecká, Tomáš Čejka

Faculty of Information Technology, CTU in Prague

*simona.buchovecka@fit.cvut.cz, tomas.cejka@fit.cvut.cz*

November 9, 2020

## Section 1

### Theory & Terminology

# What is Active Defense? Offensive Countermeasures.

- Employ offensive techniques, but with defensive posture
- Think poison (taken then consumed), not venom (injected)
- Lay traps inside your systems, but don't attack theirs
- **Does not replace the traditional layered security!**

# Current Strategies are (sometimes) Not Working

- New attack vectors
- New attack methods
- More complex environment => more difficult detection
- No silver bullet

# Cyber Deception

- Deliberate and calculated process of deceiving attackers to achieve more efficient defense
- Slow the attacker down, confuse the attacker
- Actively obfuscate your network to increase the amount of effort required to attack & noise created by attacker
- $\text{Time}(\text{Detection} + \text{Reaction}) < \text{Time}(\text{Attack})$

# Legal Issues

- Annoyance and Attribution usually pose no risk
- Attack – ALWAYS consult with legal department
- Do not hide or obfuscate what you are doing
- Use warning banners & terms of use
  - Helps to define boundaries of the network

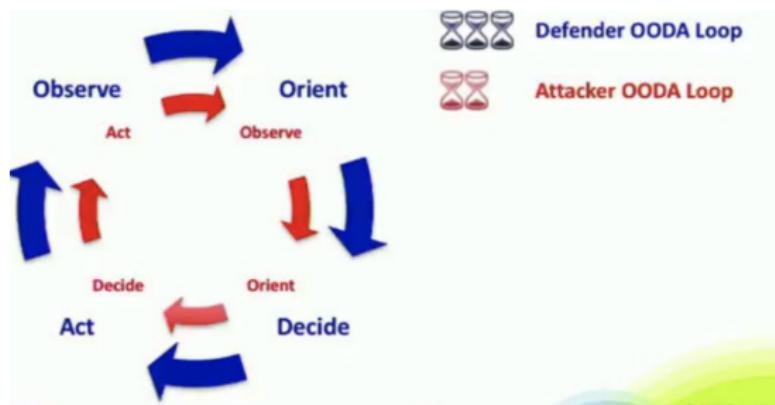
# Phases

- **Annoyance** – waste attacker's time
- **Attribution** – know who is attacking you
- **Attack** – run active code on an attacker's system

# Annoyance

- Making the system/network more difficult to attack
  - the attacker has to take more actions
  - it is more likely to detect the attack

# OODA Loop

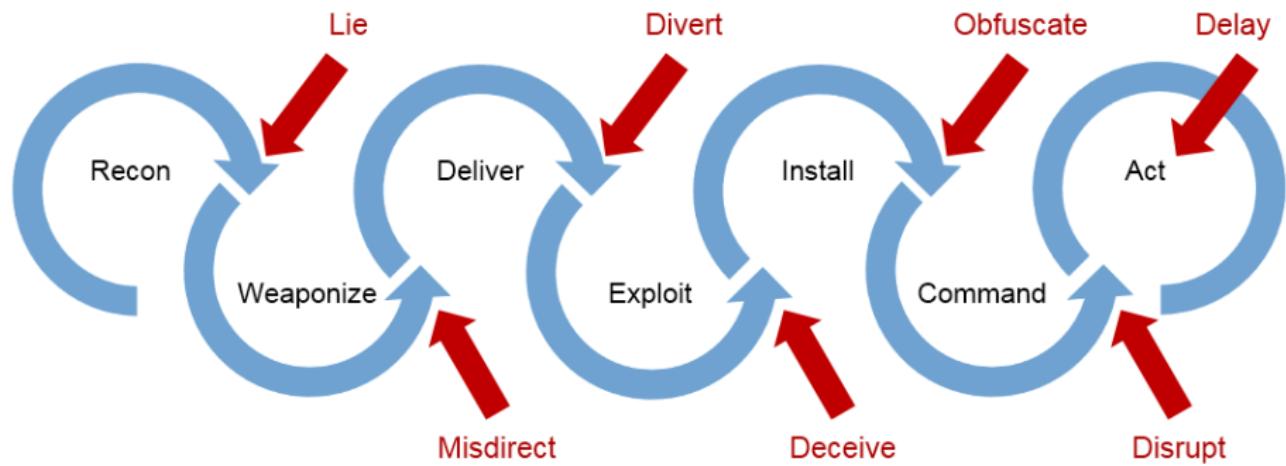


- Observe
- Orient
- Decide
- Act
- Set up tricks and traps, than watch very closely
- Whoever's OODA loop is faster, wins
- **Goal:** Disrupt attacker's OODA loop

## Phases of the Intrusion Kill Chain



# Disrupting the Kill Chain



# Obfuscate the Environment

- Change web server identification
- Change TCP/IP protocol stack of your OS
- Filter out the User-agent strings that attackers/testers use

# Spider/Crawler Traps

- Create random links and serve them for crawlers
- Be Careful – don't do it on externally facing webserver, crawled by Google – setup robots.txt (attackers may be interested into it)
- Possibly even setting up a DoS on the automated scanner
- Infinitely recursive directories

# Honeypots

## General Information

- Object intended to be interacted with by an attacker
- Research vs. Production honeypots
- Helps you to learn about attacker

## Principle

- No production system should interact with honeypot
- Any interaction with honeypot is considered malicious and should be responded to immediately
- May be interconnected into honeynet

# Honeytables

- Table within database populated with bogus data

# Honeyports

- Used to dynamically blacklist attacking systems
- Dynamic blacklisting can be source of potential issues (spoofing originating system may cause blacklisting of legitimate system)
  - Potential solution: trigger blacklisting script only when full established connection is made

- Jon Oberheide, Manish Karir, Z. Morley Mao: **Characterizing dark DNS behavior**

# Client-side Honeypot

- Automate analysis without endangering your system
- Automatically downloads and analyzes requested webpage/file

# Entrapment vs. Enticement

- Entrapment – we persuade the attacker to commit a crime that he would otherwise not commit
- Enticement – the attacker would have committed (or was intended to commit) a crime anyway – Honeypots
- Entrapment is always illegal!

# Whitelisting/System Integrity Checks

- Monitor filesystem - look for indicators of change

# Attribution

- Identify who is attacking, even if using proxies/tor . . .
- Track your intellectual property
- Google, shodan.io, censys.io, ...
- DNS tools (dig)
- Port scans, vulnerability discovery (nmap, wpscan, ...)
- Credential harvesting
- Location – geotagged media

# Dealing with Proxies/TOR

- Some of the attacker's application are configured to communicate via proxy, some of them may be not
- Goal: invoke application that might not go through the proxy and have the attacker connected to you  
→ you will get real IP address
- Application that may be used: Office, Flash, Java, ...
- Web bugs – insert arbitrary code into the file (works even if macros are disabled in Office documents) – HTML code is inserted into the document, and Word will call back

# Attack

- Once again you want to make sure that any of the techniques covered here are discussed with legal and approved by management
- Client-side attacks predominantly
- We want attacker to come to us (preferably after reviewing the warning banner)
- Even though attacker is violating law, he still has rights that you cannot violate!

# Client-side Attacks

- Java Script (e.g., BeEF framework)
- Java applets (e.g., SET)
- Macros in documents

## Section 2

### Practical Section

# Firewalls

- Stateless
- Statefull
- Port-knocking
- Application Layer Firewall

# Blacklisting

- Whitelist / greylist / blacklist
- Fail2ban

# DDoS Defense / Mitigation

- Load balancing, DNS Fluxing
- Cloud services / hosting
- CloudFlare, Google Shield
- Rate-limiting
- Remotely Triggered Blackholing (RTBH)
  - Blackhole routing
  - DNS sinkhole

# DDoS Mitigation

- Scrubbing center (appliance)
  - Corsa, A10, CESNET
- Service
  - RadWare, Akamai, CloudFlare, ...
  - [http://www.toptenreviews.com/business/internet/  
best-ddos-protection-services/](http://www.toptenreviews.com/business/internet/best-ddos-protection-services/)
- FENIX
  - Trusted secured community in the Czech peering center, can be isolated

# References

- John Strand: Offensive Countermeasures, The art of active defense
- Gartner study: Emerging Technology Analysis: Deception Techniques and Technologies Create Security Technology Business Opportunities
- Jon Oberheide, Manish Karir, Z. Morley Mao: Characterizing dark DNS behavior

# Questions?

# 7. Data Mining Techniques

Simona Buchovecká, Tomáš Čejka

Faculty of Information Technology, CTU in Prague

*simona.buchovecka@fit.cvut.cz, tomas.cejka@fit.cvut.cz*

November 9, 2020

# Data Mining Fundamentals

## Definition

Data mining is the process of automatically discovering useful information in large repositories. Data mining techniques are deployed to scour large databases in order to find novel and useful patterns that might otherwise remain unknown.

**Knowledge Discovery in Databases (KDD)** — process of converting raw data into useful information or knowledge.

Data mining is a step in the KDD process which includes::

- data preparation,
- data selection,
- data cleaning,
- incorporation of appropriate prior knowledge,
- and proper interpretation of the results of mining to ensure useful knowledge is derived from the data.

Bhuyan, Monowar H., Dhruba K. Bhattacharyya, and Jugal K. Kalita. *Network traffic anomaly detection and prevention: concepts, techniques, and tools*. Springer, 2017.

# Data Mining Tasks

## Categories

- predictive (inference on the current data in order to make future predictions)
- descriptive (characterizes the general properties of the data and underlying relationships among them)

## Most important tasks

- ① Classification and Regression
- ② Cluster Analysis
- ③ Association Analysis (discovering the most important and most strongly associated feature patterns in data)
- ④ Evolution Analysis
- ⑤ Outlier Detection

Bhuyan, Monowar H., Dhruba K. Bhattacharyya, and Jugal K. Kalita. *Network traffic anomaly detection and prevention: concepts, techniques, and tools*. Springer, 2017.

# Data Mining in Network Security #1

Do not use signatures, but learn about usage patterns

- Also viewed as Machine Learning techniques
- or Behavioral Analysis

Two basic detection approaches:

- **Misuse detection:**
  - detection of normal vs. intrusive data flows
- **Anomaly detection:**
  - classification algorithms, rare class predictive models, association rules, cost sensitive modeling

# Data Mining in Network Security #2

## Misuse Detection

- Models of misuse are created automatically
- Often better rules than manually created signatures
- Very good for detection of known (or modified) attacks
- But deciding normal vs. intrusive is human resource intensive

## Anomaly detection

- Models of normal behavior are created automatically
- Deviations from normal behavior are detected automatically
- Can potentially detect unexpected and unknown attacks

# Data Mining for Anomaly Detection

## Advantages

- Detection of unknown attacks
- Detection of unusual behavior interesting to IT managers

## Limitations

- Possibly high false alarm rate (FAR)

## Types of “training”

- **Supervised Detection**
  - Models for normal behavior are built from training data
- **Unsupervised Detection**
  - No training data, attempts to learn about anomalies automatically

# Unsupervised Anomaly Detection

## No training data

- Attempts to learn about anomalies automatically

## Algorithms used

- Statistical methods, clustering, outlier detection, state machines

# Detection Based on Association Rules

## General Information

- conceptually a simple method based on counting of co-occurrences of items in transactions databases
- unsupervised
- used for one-class anomaly detection by generating rules from data

M. V. Mahoney and P. K. Chan, *Learning rules for anomaly detection of hostile network traffic.* In Proc. 3rd IEEE International Conference on Data Mining. Washington: IEEE CS, 2003.

R. Agrawal and R. Srikant, *Fast Algorithms for Mining Association Rules in Large Databases.* In Proc. 20th International Conference on Very Large Data Bases. San Francisco, CA, USA: Morgan Kaufmann, 1994, pp. 487–499.

# Summarizing Anomalous Connections Using Association Rules

## Association patterns

- specified as frequent item sets or association rules

## Summary of detected anomalous flows:x

- Humans can analyze the summary
- E.g., a frequent set for scanning:  $srcip = X, dstport = Y$
- This is a candidate signature for a signature-based system

## Constructing a profile of normal network traffic:

- Identify sets of features that are found in normal traffic
- Examples:
  - web browsing (HTTP request): protocol=TCP, dstPort=80, NumPackets=3. . . 6
  - if port=80 and word3=HTTP/1.0 then word1=GET or POST

# Challenges in Mining Association Rules

The most difficult and dominating part of an association rules discovery algorithm is to find the itemsets that have strong support.

- an algorithm known as LERAD by Mahoney and Chan

## Challenges

- Imbalanced class distribution
  - Small support for attack, large for normal traffic
- Binarization and grouping of attribute values
  - Supervised or unsupervised
- Pruning the redundant patterns
  - Must be subsets with similar support
- Finding discriminating patterns
  - Patterns that identify attacks and normal traffic
- Grouping the discovered patterns

# Minnesota Intrusion Detection System #1

Abbreviated as MINDS Uses data mining techniques for

- unsupervised anomaly detection (assigns an “anomaly” score to each network connection)
- association pattern analysis (for suspected anomalous network connections)

Performance

- Detects new intrusions that escape signature-based IDSs

# Minnesota Intrusion Detection System #2

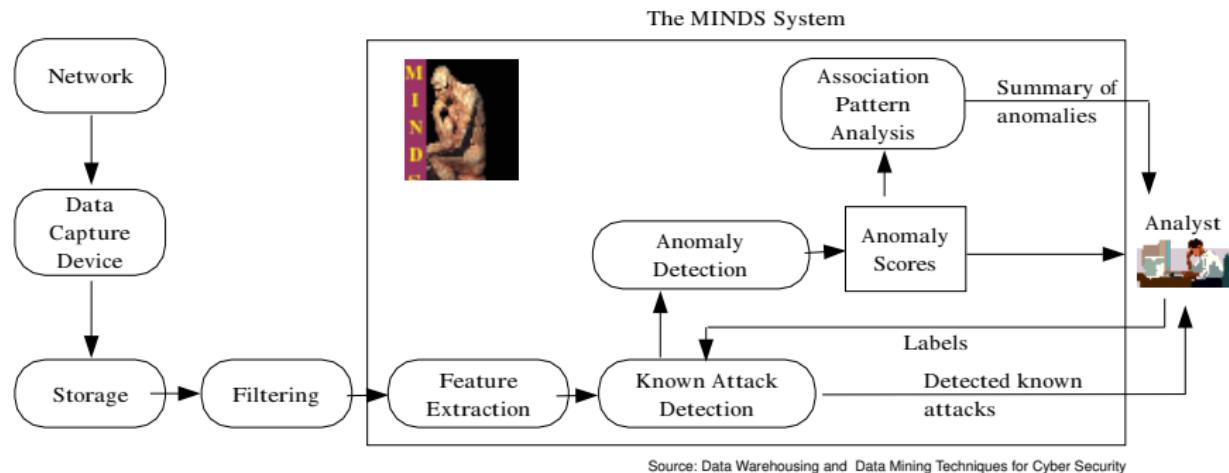
## Data capture

- Netflow v5 (Cisco protocol, today v9; IPIFIX is standard)
- Using flow-tools (<https://code.google.com/p/flow-tools/>)
- Header information only, summarized into (one-way) flows
- Much less data storage than tcpdump (or, e.g., Wireshark)
- Flow analysis period: 10 min. (1–2 mil. flows) [today 5 min.]

## MINDS processing speed

- Processing of 10 min. data: less than 3 min
- But unwanted traffic is filtered out before processing

# MINDS Architecture



# MINDS Operation Steps #1

## 1. Feature Extraction

- Basic features: source and destination IP addresses and ports, protocol, flags, number of bytes, number of packets
- Derived features for a time-window of last  $T$  seconds:
  - Capture connections with similar recent characteristics
  - Useful to detect sources of high volume connections per unit time (e.g., fast scanning)
- Derived features for a window of last  $N$  connections:
  - Similar characteristics, but for the last connections from distinct sources (good for, e.g., slow scanning)

## Time-window based features

**count-dest** Number of flows to unique destination IP addresses inside the network in the last T seconds from the same source

**count-src** Number of flows from unique source IP addresses inside the network in the last T seconds to the same destination

**count-serv-src** Number of flows from the source IP to the same destination port in the last T seconds

**count-serv-dest** Number of flows to the destination IP address using same source port in the last T seconds

## Connection-window based features

- count-dest-conn** Number of flows to unique destination IP addresses inside the network in the last N flows from the same source
- count-src-conn** Number of flows from unique source IP addresses inside the network in the last N flows to the same destination
- count-serv-src-conn** Number of flows from the source IP to the same destination port in the last N flows
- count-serv-dest-conn** Number of flows to the destination IP address using same source port in the last N flows

# MINDS Operation Steps #2

## 2. Signature based detection for known attacks

- Detected attacks are not further analyzed. Not our focus now.

## 3. Anomaly detection

- Outlier detection assigns an anomaly score to network flows
- Humans can analyze only the most anomalous connections

## 4. Association pattern analysis

- Summarization of highly anomalous network connections
- Humans decide whether the summaries can be used to create signatures for detection in step 2

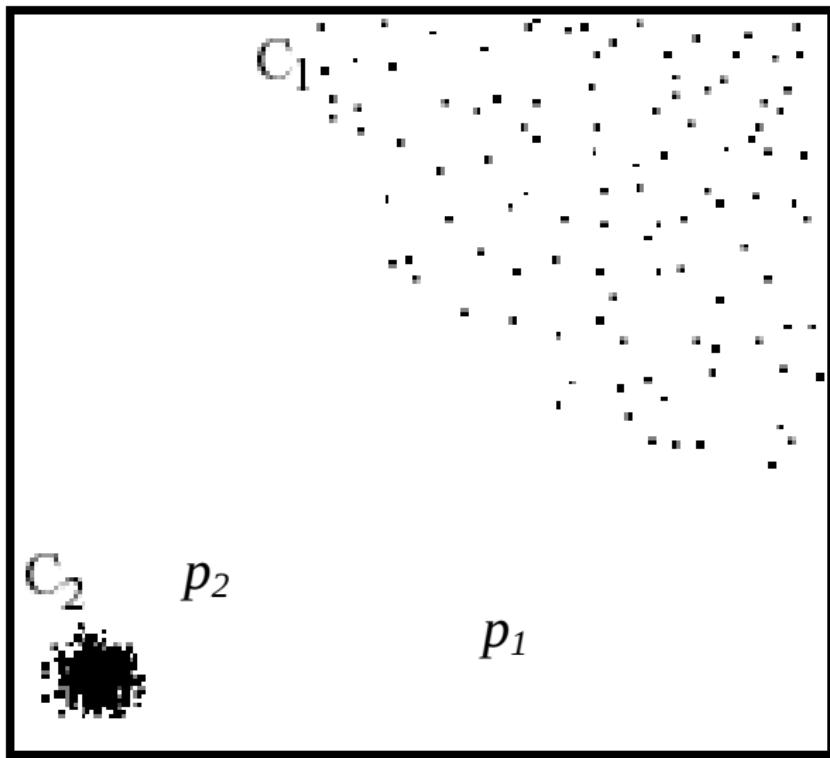
## Local Outlier Factor (LOF)

- Assigned to each data point
- It is local: outliers with respect to their neighborhood

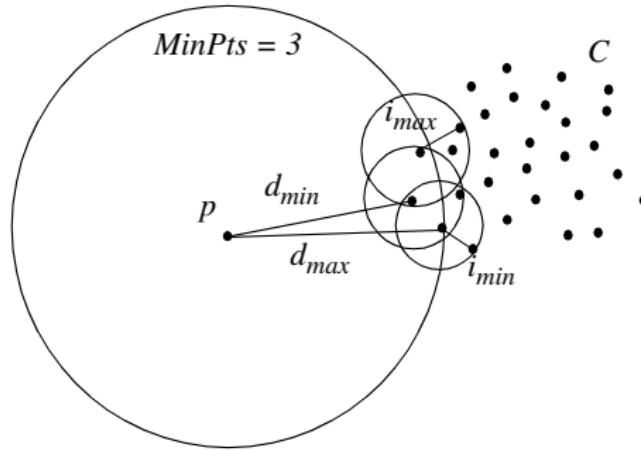
## LOF Calculation

- Compute density of each point's neighborhood
- Find the average of the density ratios over all its neighbors
- Calculate the ratio of the point's density and the density of all its neighbors

# 2D Outlier Detection



# LOF Bounds



$$d_{min} = 4 * i_{max}$$

$$\Rightarrow LOF_{MinPts}(p) \geq 4$$

$$d_{max} = 6 * i_{min}$$

$$\Rightarrow LOF_{MinPts}(p) \leq 6$$

More details in:

Breunig, Markus M., et al. *LOF: identifying density-based local outliers.* ACM sigmod record. Vol. 29. No. 2. ACM, 2000.

# Linear and Non-Linear Classification

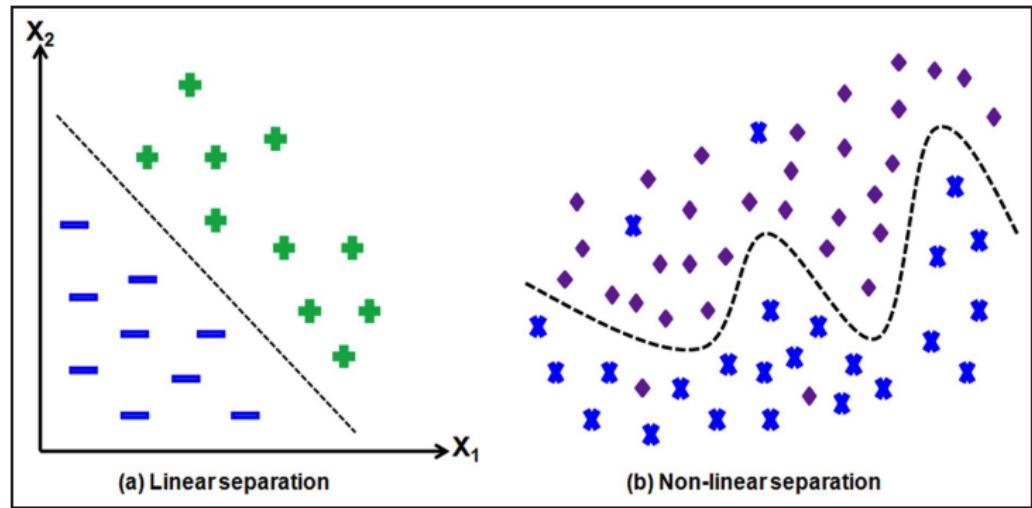
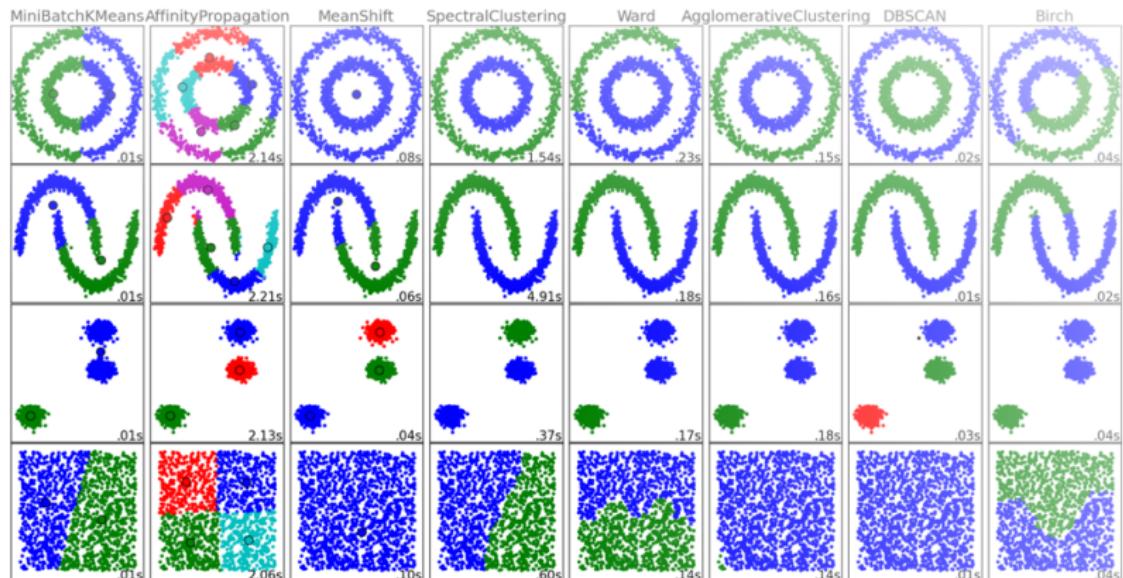


Fig. 7. Linear and non-linear classification in 2-D

Bhuyan, Monowar H., Dhruba Kumar Bhattacharyya, and Jugal K. Kalita. *Network anomaly detection: methods, systems and tools*. IEEE Communications Surveys & Tutorials 16.1 (2014): 303-336.

# Comparison of Classification Algorithms



Source: <http://scikit-learn.org/stable/modules/clustering.html>

# Computational Complexity

Neighborhoods around all data points:

- Calculate distances between all pairs of data points
- This is  $O(n^2)$  calculation – computationally infeasible for millions of data points.

Sampling a training data set from the observed data

- Compare all data points to this small set,
- Reduced complexity:  $O(\text{data size} * \text{sample size})$
- Additional benefit: anomalous behavior will not be seen often in the sample, so it will not be mistaken with normal behavior

# Performance of MINDS #1

## Worms

- October 10, 2002: MINDS detected two activities of the slapper worm that were not identified by SNORT since they were new variations of an existing worm code

## Scanning and DoS activities

- August 9, 2002: CERT/CC issued an alert for “widespread scanning and possible denial of service activity targeted at the Microsoft-DS service on port 445/TCP” August 13, 2002: Network connections related to such scanning were the top ranked outliers in MINDS The port scan module of SNORT failed (slow scanning)

# Performance of MINDS #2

## Policy Violations

- August 8 and 10, 2002: MINDS detected a machine running a Microsoft PPTP VPN server, and another one running a FTP server on non-standard ports.
- Both policy violations were the top ranked outliers since they are not allowed, and therefore very rare.
- February 6, 2003: unsolicited ICMP echo reply messages to a computer previously infected with Stacheldract worm (a DDoS agent) were detected by MINDS.
- The infected machine has been removed from the network, but other infected machines outside the local network were still trying to talk to the previously infected machine.

# Questions?

## 8. Statistical and Sequential Intrusion Detection

Simona Buchovecká, Tomáš Čejka

Faculty of Information Technology, CTU in Prague

*simona.buchovecka@fit.cvut.cz, tomas.cejka@fit.cvut.cz*

November 19, 2018

# Introduction

- Based on network traffic (statistical) distribution
- The simplest way to build a statistical model: to compute the parameters of a probability density function for each known class of network traffic and test an unknown sample to determine which class it belongs to.
- Parametric vs Non-Parametric:
  - Parametric assume knowledge of underlying distribution and estimate parameters from the given data
  - Non-Parametric do not generally assume knowledge of the underlying distribution
- Batch vs Sequential processing:
  - Batch process independent time slots
  - Sequential process a data stream, time series

# Non-statistical approach

Non-statistical features of network intrusions:

- Network protocols are deterministic and well understood
- Protocol anomalies can be detected by stateful analysis
- Many ad-hoc methods work well to detect various attacks

Example of a good deterministic ad-hoc detection rule

- Suspect a host is using P2P transfers if it:
  - uses network flows via port 6881
  - uses ports above 50000, with many port changes
  - connects to many IPs, most of them inaccessible
  - at the end many connection finish at the same time

# Statistical Approach

Statistical features of network intrusions:

- Network intrusions occur randomly
- Intrusions occur at unknown points in time
- Intrusions lead to changes of statistical properties of some observable characteristics

Attack detection viewed as a change-point detection (CPD):

- Detect changes in the distributions (models, parameters)
- With fixed delays (batch-sequential approach)
- Or with minimal average delays (sequential approach)
- While maintaining the false alarm rate at a given level

# Section 1

## Change-Point Detection

# Change-Point Detection Methodology

Observed sequence of random variables (or vectors):  $X_1, X_2, \dots$

- $X_1, X_2, \dots, X_n$  represent some network characteristics observed at times  $t_1, t_2, \dots$
- Examples: numbers of deauthentication frames, numbers of failed connections, levels of link saturation etc.

A change in distribution occurs at an unknown index  $\lambda$

- The change corresponds to a network traffic anomaly at time  $t_\lambda$
- $P_k$  and  $E_k$  denote the probability and the expectation when  $\lambda = k$
- $P_0$  and  $E_0$  correspond to the pre-change and the no-change distribution

# CPD Methodology

Sequential CPD procedure:

- Stopping time  $\tau$  is the time of alarm  
(i.e., detection of a distribution change)
- Detection delay:  $\text{ADD}_\lambda(\tau) = \mathbf{E}_\lambda(\tau - \lambda | \tau \geq \lambda)$
- False alarm rate:  $\text{FAR}(\tau) = \frac{1}{\mathbf{E}_0(\tau)}$

# Classical Optimal CPD Procedures

Conditional probability density function (pdf) for  $X_1, X_2, \dots$ :

- Before change ( $n < \lambda$ ):  $p_0(X_n | X_1, \dots, X_{n-1})$  (baseline distribution)
- After change ( $n \geq \lambda$ ):  $p_1(X_n | X_1, \dots, X_{n-1})$  ("attack" scenario)

Log-likelihood ratio (LLR):

$$Z_{n,\lambda} = \sum_{k=\lambda}^n \log \frac{p_1(X_k | X_1, \dots, X_{k-1})}{p_0(X_k | X_1, \dots, X_{k-1})}$$

Classical Change-Point Detection

- Uses a threshold of a statistic based on LLR
- Detection when the statistic first exceeds a given threshold
- It is in fact testing hypotheses that the change occurred at the point  $\lambda$  versus that there is no change at all ( $\lambda = \infty$ )

# Classical Optimal CPD Procedures

Shiryayev-Roberts-Pollak (SR) procedure:

- Motivated by Bayesian considerations
- “Average LLR” Statistic:

$$R_n = \sum_{\lambda=1}^n \exp\{Z_{n,\lambda}\}$$

- Stopping time:

$$\tau_{SP}(h) = \min\{n \geq 1 : \log R_n \geq h\}$$

Page's cumulative sum (CUSUM) procedure

- Motivated by a maximum likelihood argument
- Maximum LLR statistic:

$$\max_{1 \leq \lambda \leq n} Z_{n,\lambda}$$

- Stopping time.

$$\tau_{CU}(h) = \min\{n \geq 1 : U_n \geq h\}$$

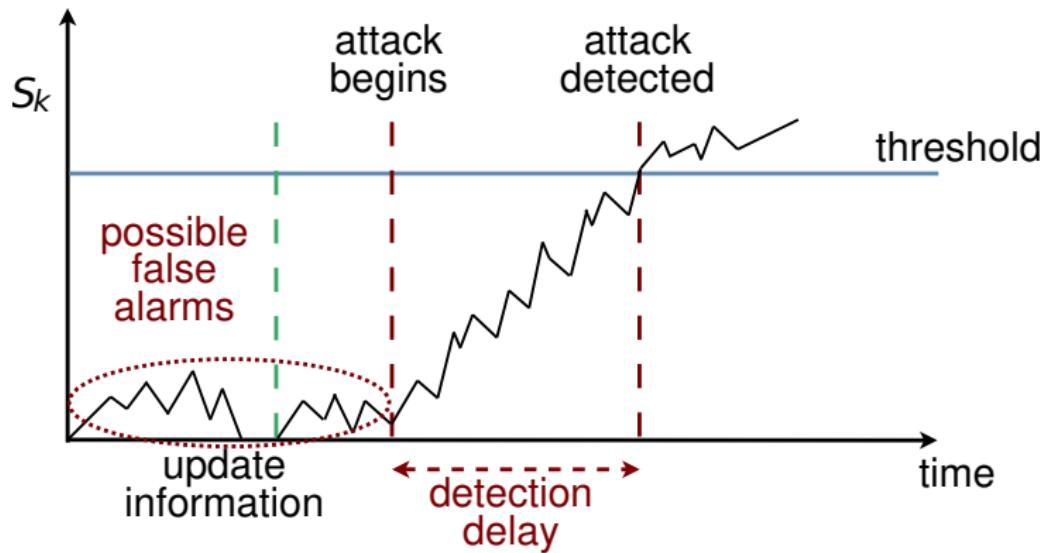
# Classical Optimal CPD Procedures

If observations are i.i.d. (independent, identically distributed)

- Both methods minimize the worst-case average detection delay  $\sup_{\lambda} \text{ADD}_{\lambda}(\tau)$
- Minimization among all methods for which the FAR is fixed ( $\text{FAR}(\tau) \leq \varphi$ )
- Thresholds should be chosen from the conditions  $E_0(\tau) = 1/\varphi$
- A preliminary threshold  $h = \log(1/\varphi)$  guarantees  $\text{FAR}(\tau) \leq \varphi$  (for both methods)
- $U_n$  can be replaced with

$$\tilde{U}_n = \max \left\{ 0, \tilde{U}_{n-1} + \log \frac{p_1(X_n)}{p_0(X_n)} \right\}$$

# CPD Example



# Generalized CPD Procedures

The i.i.d. assumption is very restrictive for intrusion detection

- Network data are usually correlated and non-stationary, even bursty, due to substantial temporal variability
- Recent CPD advances: CUSUM and SR are also optimal for general statistical models when the FAR is low ( $\varphi$  is small).

Classical optimal CPD methods require complete prior knowledge of the pre-change and post-change distributions

# Generalized CPD Procedures

Parametric modifications:

- Generalized likelihood ratio (LR), LR mixtures, and adaptive LR
- Do not solve the problem when the distributions are not known
- Procedures based on signs or ranks are not quite computationally efficient

Non-Parametric NP-CUSUM procedure (by Tartakovsky et al.):

- Inspired by the CUSUM statistic  $\tilde{U}_n = \max \left\{ 0, \tilde{U}_{n-1} + \log \frac{p_1(X_n)}{p_0(X_n)} \right\}$
- Asymptotically optimal: ADD is nearly minimized for low FAR
- Has manageable computational complexity

# Generalized Non-Parametric Sequential Statistical Detection

$$S_n = \max \{0, S_{n-1} + f_n(X_n)\}, S_0 = 0$$

If we knew the exact probabilistic models for the pre- and post-attack scenarios, then  $f_n(X_n)$  would be the Log-Likelihood Ratio.

# Non-Parametric Sequential Statistical Learning

$$S_n = \max \left\{ 0, S_{n-1} + X_n - \mu - \varepsilon \hat{\theta}_n \right\}, S_0 = 0,$$

where:

- $X_n$  is an observed network characteristic in the  $n^{\text{th}}$  time interval (e.g., number of UDP, TCP SYN, ICMP, or ARP packets),
- $\mu$  is a historical estimate of  $E(X_n)$ ,
- $\varepsilon$  is a tuning parameter,
- $\hat{\theta}_n$  is an estimate of  $E(X_n)$  under attack.

## Section 2

### Smoothing and Predicting

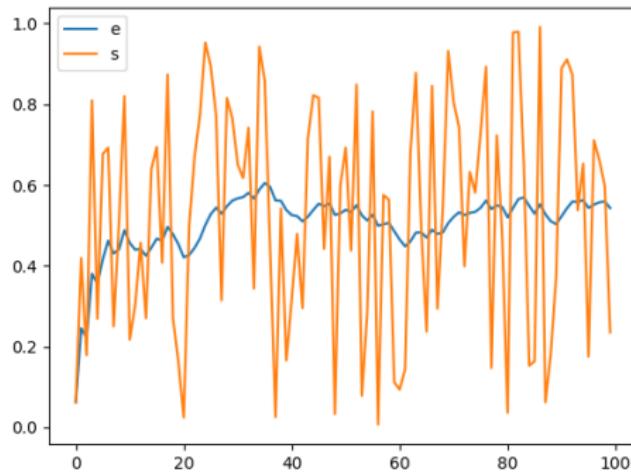
# Exponential Weighted Moving Average (EWMA)

- Short-term peaks might produce false alerts
- Smoothing of the observed characteristic

$$z_i = \alpha x_i + (1 - \alpha)z_{i-1}, z_0 = x_0,$$

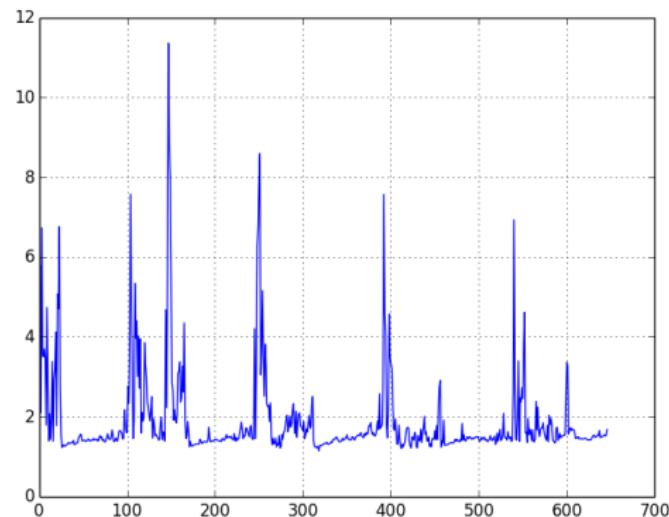
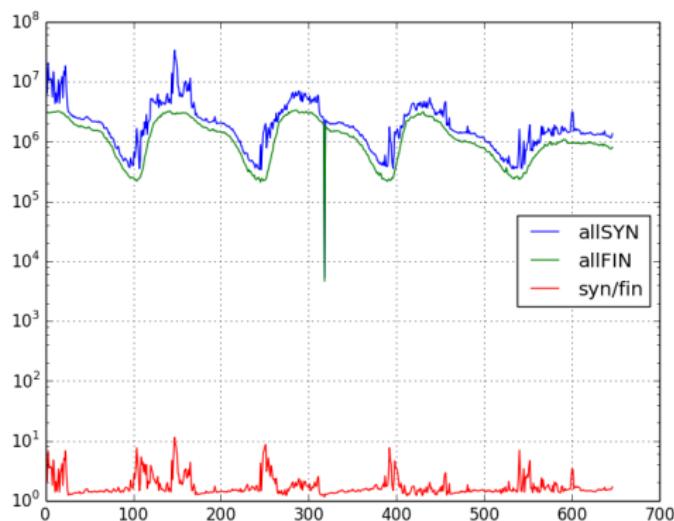
where  $\alpha$  is the coefficient of smoothing.

Example with  $\alpha = 0.05$ :



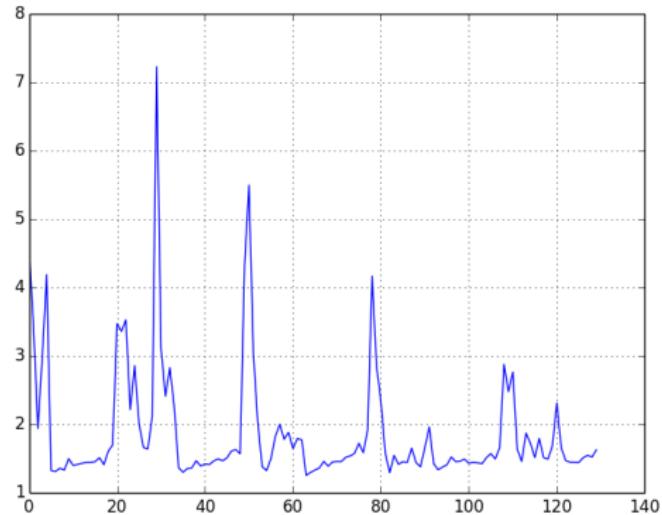
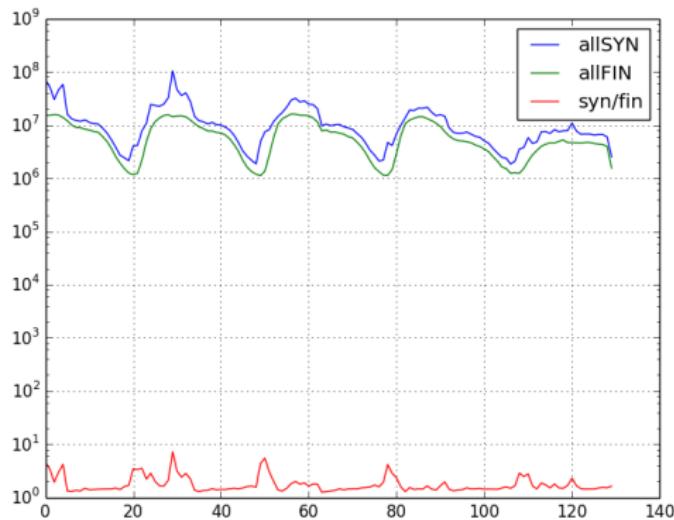
# Smoothing effect of aggregation 1 min

Observed number and SYN/FIN ratio



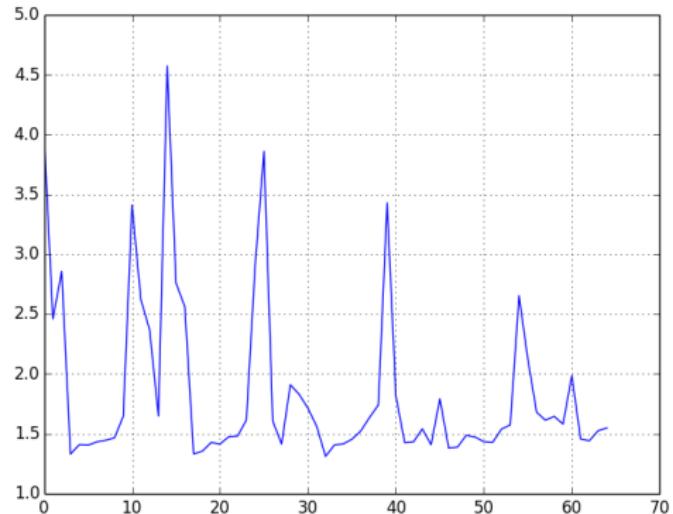
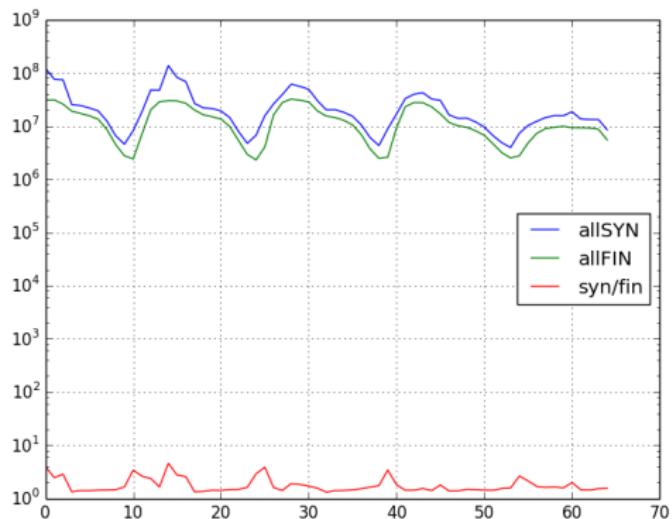
# Smoothing effect of aggregation 5 min

Observed number and SYN/FIN ratio



# Smoothing effect of aggregation 10 min

Observed number and SYN/FIN ratio



# Holt-Winters — Prediction and Seasonality

$$\hat{y}_{i+1} = a_i + b_i + c_{i+1-m} \quad (1)$$

where  $a$ ,  $b$ ,  $c$  are components defined as:

$$\begin{aligned} a_i &= \alpha(y_i - c_{i-m}) + (1 - \alpha)(a_{i-1} + b_{i-1}) \\ b_i &= \beta(a_i - a_{i-1}) + (1 - \beta)b_{i-1} \\ c_i &= \gamma(y_i - a_i) + (1 - \gamma)c_{i-m} \end{aligned} \quad (2)$$

$a_i$  “baseline” or “intercept”,

$b_i$  “linear trend” or “slope”,

$c_i$  “seasonal trend”,

$\alpha, \beta, \gamma$  adaptation parameters,  $0 < \alpha, \beta, \gamma < 1$ .

The detection is based on a confidence band  $(\hat{y}_i - \delta_- d_{i-m}, \hat{y}_i + \delta_+ d_{i-m})$  where  $\hat{y}_i$  is a predicted value and  $\delta_-, \delta_+$  are scaling factors for the width of the confidence band.

# Future Prediction

- Currently, scope of Machine Learning
- Having historical data, a predictor can be trained
- Random forests, XGBoost (Extreme Gradient Boosting), Neural Networks, ...

## Section 3

### Common Performance Metrics

## Test Power and Probability of False Alert

The performance of the sequential detection procedures can be measured using various of criteria: Test power (PWR) and Probability of False Alert (PFA):

- We desire high power and low probability of false alerts.
- For a fixed decision time  $k$  we can define:

$$\text{PWR}^k = P(\tau \leq k | \lambda \leq k)$$

$$\text{PFA}^k = P(\tau \leq k | \lambda > k) = P_0(\tau \leq k)$$

- In long-term network monitoring the change-point  $\lambda$  (i.e., an intrusion) may occur very late.
- For large  $k$ , any detection procedure will have  $\text{PFA}^k$  nearly 1 or at least relatively high.

# PWR and PFA in practice

- It is more practical to consider conditional PFA for a sliding time interval of length  $T$

$$\text{PFA}_T^k = P(\tau < k + T | \tau \geq k, \lambda \geq k + T) = P_0(\tau < k + T | \tau \geq k)$$

$$\text{PFA}_T = \sup_{1 \leq k \leq \infty} \{P_0(\tau < k + T | \tau \geq k)\}$$

- The condition  $\tau \geq k$  corresponds to false alarms shortly after (within) period  $T$ 
  - The IDS was inspected and found OK at time  $k$
  - The IDS was started or restarted after an alert ( $k = 0$ )
  - An upper bound on  $\text{PFA}_T$  works for all of these situations

# Run Length and False Alert Rate

- The average run length before the change

$$ARL^0 = E_0\tau$$

- The average run length after the change

$$ARL^1 = E_1\tau$$

- We desire quick detection (low  $ARL^1$ ) and infrequent false alerts (high  $ARL_0$ )
- The average False Alert Rate

$$FAR(\tau) = \frac{1}{E_0\tau}$$

is often used instead of  $ARL_0$

- Low FAR is a very important and practical requirement!

# Detection Delay

- Average detection delay, assuming a change at a fixed  $\lambda = k$

$$\mathbf{E}_k(\tau - k)^+$$

- As  $k$  increases, the delay often approaches 0.
- For a random  $\lambda$  we can summarize:  $\mathbf{E}(\mathbf{E}_\lambda(\tau - k)^+)$
- Conditional average detection delay

$$\text{ADD}_\lambda(\tau) = \mathbf{E}_\lambda(\tau - \lambda | \tau \geq \lambda)$$

- Very important in continual surveillance.
- Often approaches a constant for increasing  $\lambda = k$  (stable detection system)

# Utility Function

- There is often cost associated with
  - False alerts and detection delays
  - Communication overhead in network security
- If the cost (utility function) is a linear function of FAR and delay, an optimal procedure can in some cases be obtained by fixing the FAR and minimizing the ADD

# Evaluation Criteria

- Accuracy: measures how correctly an IDS works, percentage of detection and failure, false alarms  
(90 % accuracy means correct classification of 90 instance of 100 as belonging to their actual classes)
- Taxonomy of evaluation measures:
  - Data Quality (Quality, Validity, Completeness, Reliability)
  - Correctness (P-R and F-measures, ROC Curves, Misclassification Rate, Confusion Matrix, AUC Area, Sensitivity and specificity)
  - Efficiency (Stability, Timeliness, Unknown Attack, Update Profile, Interoperability, Performance, Generate Alert)

## Data Quality

**Quality** reliability/legitimacy of source, good selection of samples (unbiased), good sample size (neither over nor under-sampling), time of data, complexity of data

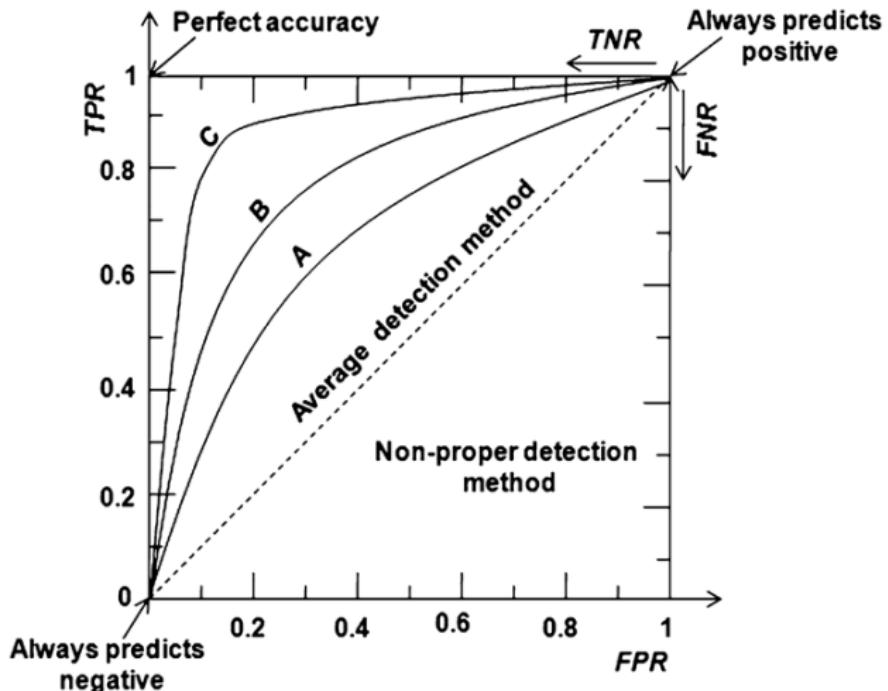
**Reliability** accuracy, consistency, expected purpose

**Validity** valid data, e.g., good values in expected ranges

**Completeness** represent the space of the vulnerabilities and attacks that can be covered by an IDS

## Correctness

**ROC Curve** Receiver Operating Characteristics, originates from signal processing



See ROC Example at <https://medium.com/wwblog/>

**AUC** Area Under Curve, computed from ROC, result is within the range 0.5–1.0

Precision, Recall, and F-Measure defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \text{TPR} = \frac{\text{TP}}{\text{Pos}} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F-measure} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Pos} + \text{Neg}}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

$$\text{TNR} = \frac{\text{TN}}{\text{Neg}} = \frac{\text{TN}}{\text{FP} + \text{TN}} = 1 - \text{FPR}$$

$$\text{FNR} = \frac{\text{FN}}{\text{Pos}} = \frac{\text{FN}}{\text{TP} + \text{FN}} = 1 - \text{TPR}$$

Confusion Matrix is composed of TP, FP, FN, TN:

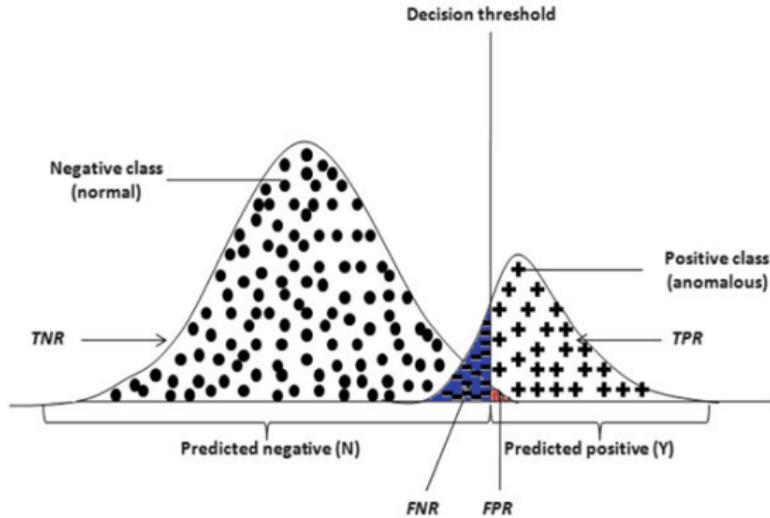
		p	True class	n
Predicted class	Y	True Positive (TP) Good: <i>Correct detection</i>	False Positive (FP) Bad: <i>Type-I error</i>	
	N	False Negative (FN) Bad: <i>Type-II error</i>	True Negative (TN) Good: <i>Correct rejection</i>	
		Pos= TP+FN (Total number of actual positives)	Neg= FP+TN (Total number of actual negatives)	
<b>CONFUSION MATRIX</b>				

Misclassification Rate

$$\frac{FN + FP}{TP + FP + FN + TN}$$

## Sensitivity and Specificity

- TPR is also known as *sensitivity*
- TNR is also known as *specificity*
- TPR, FPR, TNR, and FNR can be defined for the normal class



## Tradeoffs in continual surveillance:

Tuning Adjustment	Effects on FAR and ADD	PFA and PWR in a time interval of given size	Effects on overall PFA
Higher Threshold	Smaller FAR Shorter ADD	Smaller PFA Lower Power	PFA approaches 1
Lower Threshold	Higher FAR Shorter ADD	Higher PFA Higher Power	PFA approaches 1
Optimization Strategy A	Limit $\text{FAR} \leq \gamma$ Minimize ADD	Limit $\text{PFA} \leq \alpha$ Maximize Power	Not applicable
Optimization Strategy B	Limit $\text{ADD} \leq K$ Minimize FAR	Guarantee Power $\geq \beta$ Minimize PDA	Not applicable

## Efficiency (related generally to detection systems)

**Stability** detection performance is consistent in different network scenarios

**Timeliness** total delay between  $t_{\text{attack}}$  and  $t_{\text{response}}$

**Performance** throughput of the detection method (e.g., how many packets/second without loss), CPU and memory usage → computational and memory complexity

**Update Profile** possibility to add new or modified profiles or signatures accurately

**Interoperability** capability to correlate information from multiple sources

**Unknown Attack** ability to detect unknown or modified intrusion patterns

# Detection Performance Metrics Tradeoffs

- Algorithms cannot maintain all metrics at prescribed levels
- Optimization of detection procedures balances the tradeoffs

A standard optimization strategy:

- Prescribe the bounds for some metrics
- Use some other metrics as optimization criteria

The selection of these metrics has practical considerations

- Classical approach: maximize the test power among all tests with a fixed prescribed low level of PFA
- Continual surveillance: minimizing the detection delay for a prescribed low FAR

## Section 4

Closing Words

- ① Bhuyan, Monowar H., Dhruba K. Bhattacharyya, and Jugal K. Kalita. *Network traffic anomaly detection and prevention: concepts, techniques, and tools*. Springer, 2017.
- ② R. Blažek, H. Kim, B. Rozovskii, and A. Tartakovsky, *A novel approach to detection of 'denial-of-service' attacks via adaptive sequential and batch- sequential change-point detection methods*, Proc. 2nd IEEE Workshop on Systems, Man, and Cybernetics, West Point, NY, 2001.
- ③ A. Tartakovsky, B. Rozovskii, R. B. Blažek, and H. Kim, *Response of authors to discussions on detection of intrusions in information systems by sequential change-point methods*, Statistical Methodology, vol. 3, pp. 329–340, July 2006.
- ④ A. Tartakovsky, B. Rozovskii, R. B. Blažek, and H. Kim, *Detection of intrusions in information systems by sequential change-point methods*, Statistical Methodology, vol. 3, pp. 252–293, July 2006.

- ⑤ B. Rozovskii, A. Tartakovsky, R. B. Blazek, and H. Kim, *A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods*, IEEE Transactions on Signal Processing, vol. 54, pp. 3372–3382, September 2006.
- ⑥ <https://www.symantec.com/connect/articles/statistical-based-intrusion-detection>
- ⑦ <https://medium.com/wwblog/evaluating-anomaly-detection-algorithms-with-receiver-operating-characteristic-curves-5a2a2a2a2a2a>
- ⑧ <https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/>

# Questions?

# 9. Network Intrusion Detection Systems

Simona Buchovecká, Tomáš Čejka

Faculty of Information Technology, CTU in Prague

*simona.buchovecka@fit.cvut.cz, tomas.cejka@fit.cvut.cz*

April 19, 2021

# Section 1

## Introduction

# Abbreviations

## IDS Intrusion Detection System

- focused on suspicious/malicious traffic detection
- host-based / network-based
- signature-based (recognition of bad patterns, malware)

## ADS Anomaly Detection System

- statistical, machine learning, or other approach to detection
- anomalous traffic — deviation from normal, usually observed
- detection of known or unknown traffic
- “behavioral analysis”

## IPS Intrusion Prevention System

- identify malicious activity, log information, report it, try to block/stop it
- packet discarding (/blackholing), connection resetting

## Section 2

### Particular Detection Systems

# Existing Tools (mainly IDS)

## Packet-based

- Snort
- Zeek (formerly Bro)
- Suricata
- ...

## Flow-based

- Flowmon ADS / DDoS Defender
- Stream4Flow
- ntopng
- NfSen (batch processing)
- Analysis Pipeline (SiLK)
- NEMEA
- ...

# Existing Tools (mainly IDS)

## Offline Processing

- Elasticsearch + Kibana
  - elastalert (<https://github.com/Yelp/elastalert>)
  - elastiflow (<https://github.com/robcowart/elastiflow>)
- Python + Pandas, matplotlib
- ...

# Snort Introduction

IDS/IPS - Intrusion Detection / Prevention System

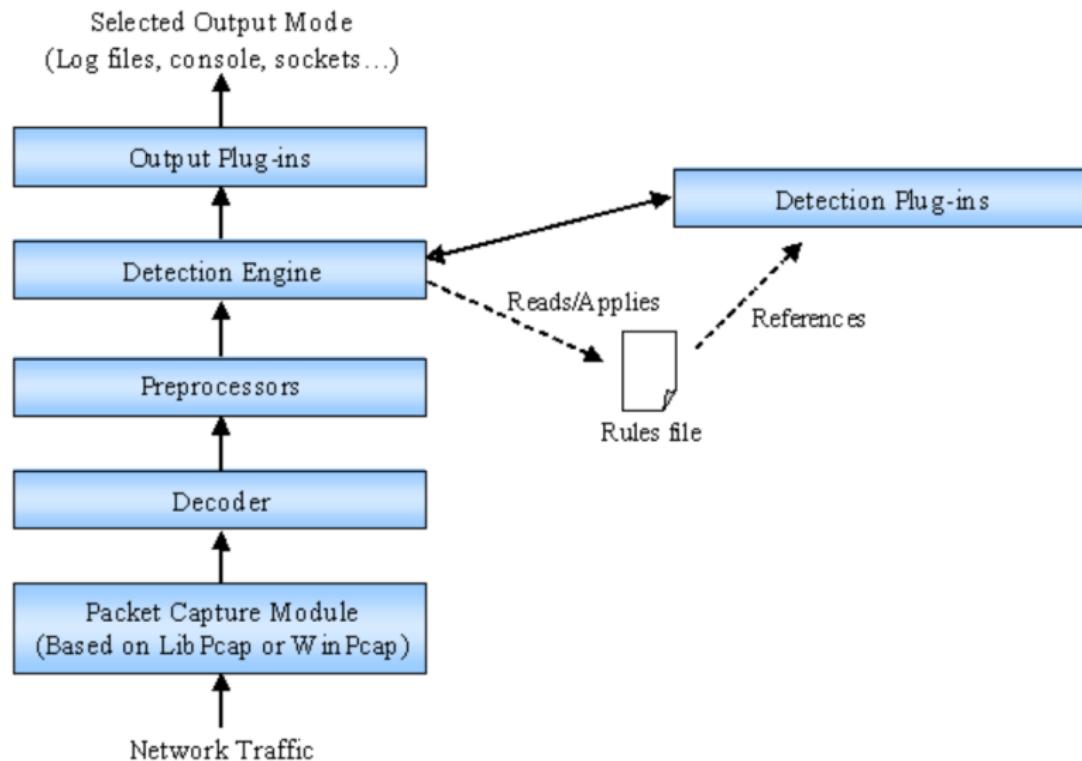
- Statefulness: Rule-based detection with thresholds to track the number of times a rule was triggered

Open source, developed by Cisco Systems (formerly SourceFire)

Combines:

- Signature, protocol, and anomaly-based inspection
- But usually it's packet-based
- Sniffer, packet logger, intrusion detection

# Snort Structure & Plugins



# Snort Components #1

## Packet Decoder

- Takes packets from various interfaces

## Preprocessors

- Arrange or modify packets
- E.g., convert unicode or hex characters in URL to text
- Reassemble fragmented IP packets
- Reassemble TCP segments
- Check for anomalies in packet headers (and issue alerts)
- Port scan processor

# Snort Components #2

## Detection Engine

- Detects intrusions in packets
- The detection engine speed is critical
- Uses Snort rules:
  - Rules arranged in a chain
  - If a rule is matched, a defined action is taken
  - Otherwise, the packet is dropped
  - *Actions:* Logging packets, issuing alerts, ...

## General principles:

- The first rule that applies generates action/alert
- The highest priority applicable rule generates action/alert

# Snort Components #3

Rules are applied to:

- IP Header
- Transport Layer Header (TCP, UDP, ICMP)
- Application layer header
- Packet payload (legal issues)

Detection engine performance depends on:

- Machine power
- Internal bus speed
- Network load
- Number of rules

# Snort Components #4

## Logging and Alerting System

- Logs in /var/log/snort by default

## Output Plugins

- Log to /var/log/snort/alerts or other file, or via syslog facility
- Log to a database (MySQL, Oracle)
- Send e-mails, show web-based alerts
- Send SNMP traps
- Generate XML output
- Modify configuration of routers or firewalls
- Send SMB messages to MS Windows machines

# Snort Rules #1

Intro: <https://resources.infosecinstitute.com/snort-rules-workshop-part-one/>

General form:

```
action proto src_ip src_port direction dst_ip  
dst_port (options)
```

Log 100 packet if ssh exploit is suspected

```
activate tcp any any -> 192.168.1.21 22  
(content:"/bin/sh"; activates:1; \  
msg:"Possible SSH buffer overflow"; )
```

```
dynamic tcp any any -> 192.168.1.21 22  
(activated_by:1; count:100;)
```

# Snort Rules #2

Custom action:

```
ruletype redalert
{
    type alert
    output alert_syslog: LOG_AUTH LOG_ALERT
    output database: log, mysql, user=snort
    dbname=snort host=localhost
}
```

# Snort Rules #3

Detect Nop:

```
alert tcp any any -> any any (msg:"Possible exploit"; \
content:"|90|"; offset:40; depth:75; dsize: >6000;)
```

SYN & FIN Sent in one packet:

```
alert any any -> any any (flags: SF,12; \
msg: "Possible SYN FIN scan";)
```

Note: Do not use escaped quotes

## Basic information

- Transforms packets into events
- Events are processed using a script interpreter
- Turing complete Bro scripting language
- Zeek analyzers, in Zeek's event engine, perform application layer decoding, anomaly detection, signature matching, connection analysis

Resources:

<https://www.zeek.org/>

<https://en.wikipedia.org/wiki/Zeek>

## Basic information

- IDS, IPS, Network Security Monitoring (NSM)
- TCP/IP engine (IPv6 support, tunnel decoding, tracking sessions, reassembling)
- protocol parsing (HTTP, SSL, TLS, SMB, ...)
- PCRE support
- Lua scripting

## Rules format

*action header options*

Example:

```
drop tcp $HOME_NET any -> $EXTERNAL_NET any
(msg:"ET TROJAN Likely Bot Nick in IRC (USA +..)";
flow:established,to_server; flowbits:isset,is_proto_irc;
content:"NICK "; pcre:"/NICK .*USA.*[0-9]{3,}/i";
reference:url,doc.emergingthreats.net/2008124;
classtype:trojan-activity; sid:2008124; rev:2;)
```

# Suricata

(3/3)



## Resources:

<https://suricata-ids.org/>

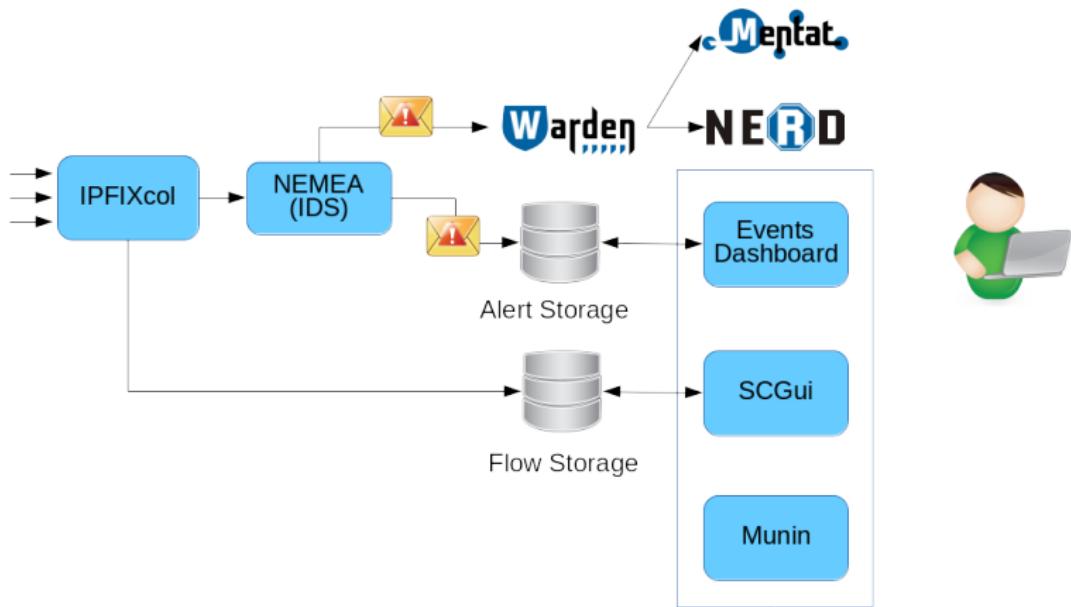
S.Buchovecka, T.Cejka (FIT, CTU)

Network Intrusion Detection Systems

April 19, 2021

19 / 30

- Modular, consisting of independent interconnected NMEA modules
- Flow-based
- Stream-wise
- Application-aware (can work with L7-extend flow records)



# NEMEA and other tools: screenshots of visualization (1/4)



## NEMEA and other tools: screenshots of visualization (2/4)

SecureCloud

Graph

Statistics

Database Query

Profiles

User Control

Selected profile: **http://192.168.1.100:8080**

Threshold: Jan 08 2017  
14:10

https://localhost:8080

Tab 1

Sources

Filter

Fast Options

Custom Options

Clear Star

Process request

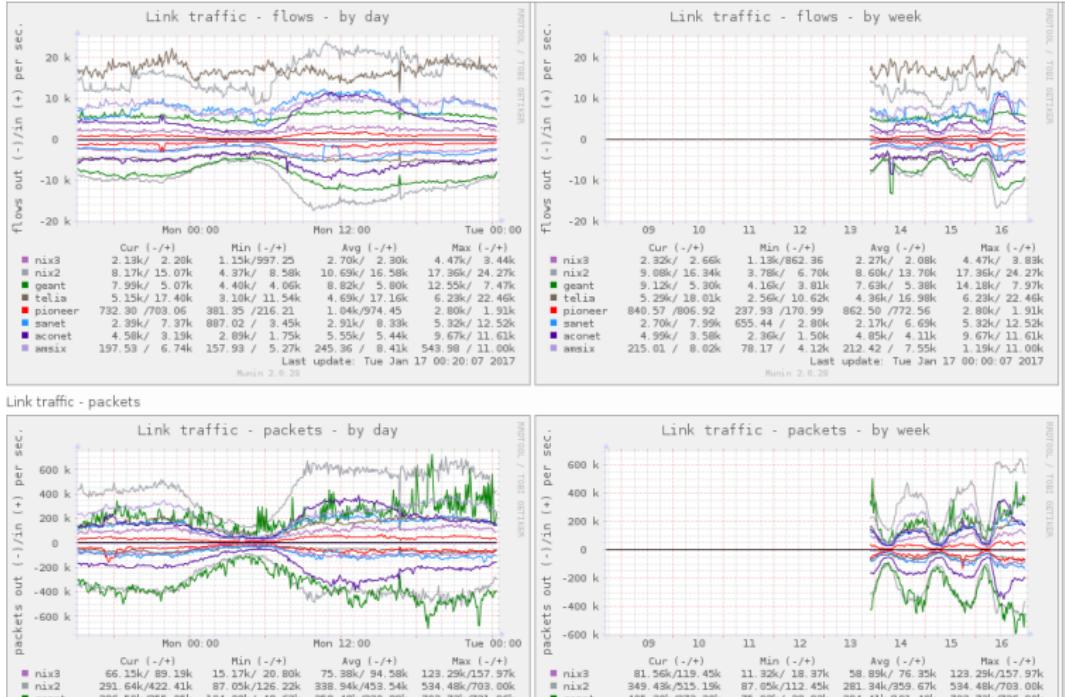
Query parameters

```
/usr/lib64/mpich/bin/mpexec -n 2 /usr/lib64/mpich/bin/fdistump_mpich -f -- -t 1400000000 -l 20 -a ssrcip,dstip -o flows --output-format=pretty --output-items=r,p
```

Query output

first	last	bytes	pkts	flows	ssrcip	dstip	duration	lat
2017-01-08 13:09:37.120	2017-01-08 13:09:52.120	260.7 k	51 k	3.5 k	60.1.12.174.184	93.113.100.53	00:00:25.000	16
2017-01-08 13:05:49.120	2017-01-08 13:07:36.120	203.4 k	35 k	3.5 k	140.150.80.37	93.113.100.50	00:01:50.000	14
2017-01-08 13:05:10.372	2017-01-08 13:08:02.120	1.3 M	85.5 k	3.5 k	70.216.1.118	93.113.173.202	00:02:51.748	85
2017-01-08 13:05:15.377	2017-01-08 13:08:02.120	76.5 M	67.5 k	2.9 k	61.49.10.61	93.113.104.195	00:02:46.749	16
2017-01-08 13:05:22.200	2017-01-08 13:08:02.120	46.4 M	2.8 k	2.8 k	179.80.143.190	93.113.249.5	00:02:30.000	26
2017-01-08 13:05:26.200	2017-01-08 13:08:02.120	256.4 k	22 k	2.8 k	179.80.143.190	93.113.249.53	00:02:30.000	16
2017-01-08 13:05:37.120	2017-01-08 13:07:30.120	3.0 M	17.4 k	2.5 k	1590.Fauf{1e1d803:ffff:ffff:ffff:ffff:ffff:ffff:ffff}fffe	dffe77e7:63:0:81a7:80881c90:f118:3b13	00:02:24.000	16
2017-01-08 13:05:36.240	2017-01-08 13:07:30.120	2.0 M	97.0 k	1.4 k	166.179.82.05	93.204.66.350	00:01:56.671	13

# NEMEA and other tools: screenshots of visualization (3/4)



# NEMEA and other tools: screenshots of visualization (4/4)



- <http://nemea.liberouter.org/>
- <http://github.com/CESNET/LiST>
- <http://github.com/CESNET/SecurityCloudGUI>

# FastNetMon

- Detects DDoS Attacks in 2 seconds
- Supports flow data, sFlow, port mirror/SPAN
- Supports BGP

```
FastNetMon v1.0
IPs ordered by: packets (use keys 'b'/'p'/'f' for change) and use 'q' for quit
Threshold is: 35000 pps and 1000 mbps total hosts: 13568

Incoming traffic      171015 pps   384 mbps  11973 flows
159.11.22.33          3309 pps    33.3 mbps   77 flows
159.11.22.33          3116 pps    34.8 mbps   2 flows
159.11.22.33          2567 pps    29.5 mbps   2 flows
159.11.22.33          2439 pps    1.8 mbps   76 flows
159.11.22.33          2364 pps    1.4 mbps   55 flows
159.11.22.33          2104 pps    1.5 mbps   19 flows
159.11.22.33          1938 pps    1.3 mbps   36 flows

Outgoing traffic     225121 pps   1905 mbps  17893 flows
159.11.22.33          3699 pps    39.9 mbps   83 flows
159.11.22.33          3557 pps    37.3 mbps  124 flows
159.11.22.33          2965 pps    32.8 mbps   98 flows
159.11.22.33          2645 pps    29.7 mbps   38 flows
159.11.22.33          2522 pps    26.1 mbps   65 flows
159.11.22.33          2474 pps    26.8 mbps   61 flows
159.11.22.33          2285 pps    18.9 mbps  194 flows

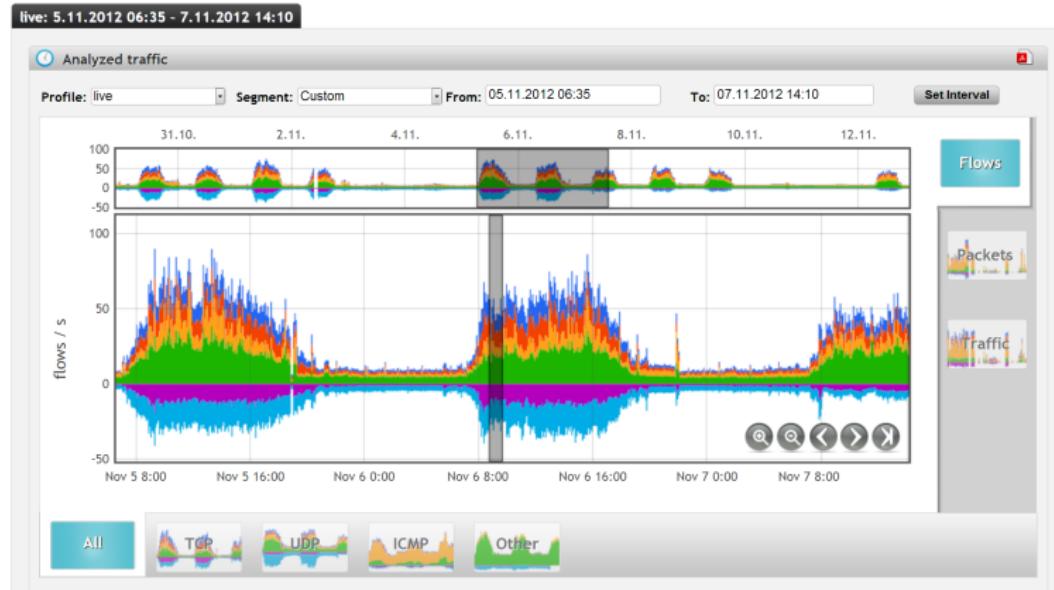
Internal traffic       0 pps     0 mbps
Other traffic          56 pps    0 mbps

Traffic calculated in: 0 sec 14670 microseconds
Packets received: 2308537
Packets dropped: 0
Packets dropped: 0.0 %
```

<https://fastnetmon.com>

# Flowmon

Commercial



<https://www.flowmon.com/en/>

## Section 3

### Closing Words

**There are many more...**

Discussion?

What are Your experiences?

# Questions?

# 10. Incident Response

Simona Buchovecká, Tomáš Čejka

Faculty of Information Technology, CTU in Prague

*simona.buchovecka@fit.cvut.cz, tomas.cejka@fit.cvut.cz*

December 2, 2019

# Event & Incident (NIST framework)

- **Event** is any observable occurrence in a system or network
- **Adverse events** are events with a negative consequence, such as system crashes, packet floods, unauthorized use of system privileges, unauthorized access to sensitive data, and execution of malware that destroys data
- **A computer security incident** is a violation or imminent threat of violation of computer security policies, acceptable use policies, or standard security practices.
- Security Incident != Operations (ITIL) incident (different objectives — security: stop the data exfiltration, minimize damage; operations: get it back to operation)

# Security incident . . . in other words. . .

- Intent to cause harm
- Performed by a person
- Involves computing resource
- Examples
  - Data theft
  - Theft of funds - bank access, credit card and wire fraud
  - Extortion
  - Unauthorized access to computing resources
  - Presence of malware
  - Possession of illegal or unauthorized materials

# What exactly is an incident response?

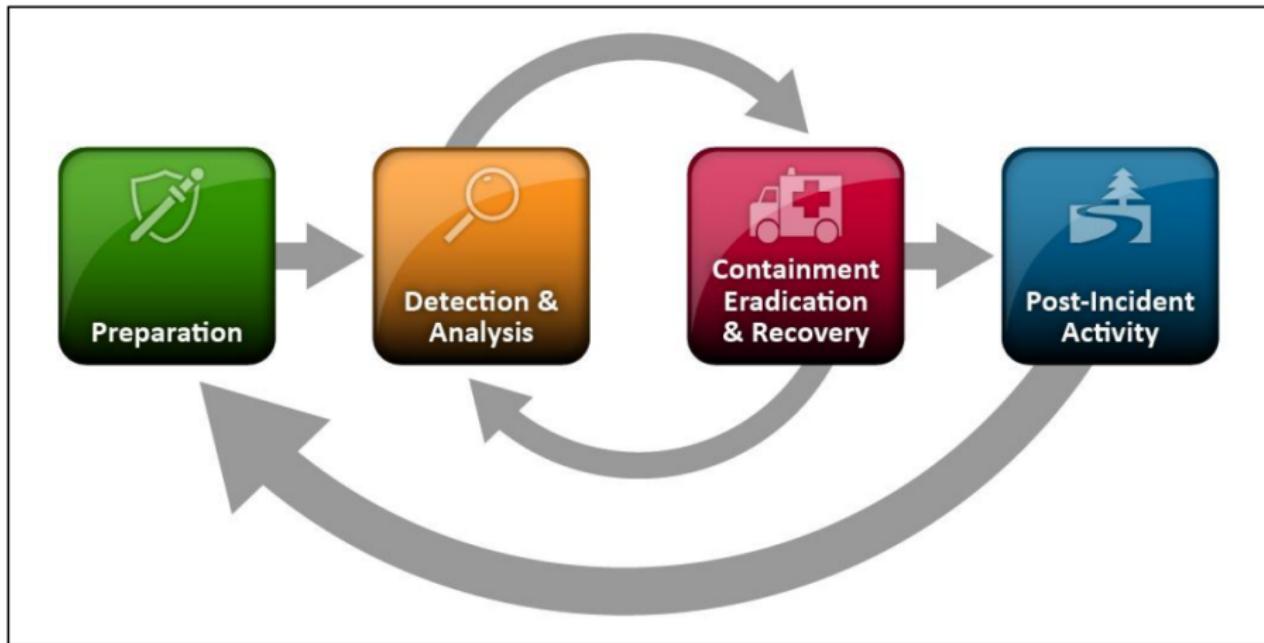
- Confirm whether or not an incident occurred
  - False positives x True positives
  - False negatives x True negatives
- Provide rapid detection & containment
  - (isolation of the threat/infection)
- Determine and document the scope of the incident
- Minimize the disruption to the business
- Minimize the damage
- Restore normal operations
- Allow for criminal or civil actions against attackers
- Educate senior management
- Enhance security posture...

# Need for Incident Response

- Incidents happen (no matter how secure environment is)
- Criminals work with little risks
- Efficient and timely response to the incident can minimize the financial or reputation damage
- If there is data leak, we need to know what was leaked
- Ensure that incident is remediated properly (and attackers are out of your network)

# Handling an Incident

*It's critical to follow the process and not skip the phases!*



# Preparation



- Prepare to handle incidents
  - Incident Response Policy, Plan, and Procedure Creation
    - Sharing information/reporting to 3rd parties (e.g., Incident involving personal data)
    - Management buy-in, set-up expectations, communication matrix, escalation paths, ...
    - Setting up dependencies: Management, Information Assurance, IT Support, Legal Department, Public Affairs and Media Relations, Human Resources, Business Continuity Planning
  - Communications and Facilities: Contact information, incident reporting mechanism, issue tracking system, war room, ...
  - Hardware and Software: digital forensic workstations and/or backup devices, laptops, blank removable media, ...
  - Resources: Documentation, network diagrams, list of critical assets, baselines, ...
- Preventing incidents - any activity that helps you prevent incidents (network security, endpoint security, user awareness)

# Incident Response vs. Business Continuity

*Ongoing incident may have serious impacts on business operations, e.g., DoS*

## Incident Response Planning

- Security-related threats to systems, networks & data
- Data confidentiality
- Non-repudiable transactions

## Business Continuity Planning

- Disaster Recovery Plan
- Continuity of Business Operations
- IRP is part of BCP and can be the first step

# Preparation, Preparing the Incident Response Team (IRT)



- Central IRT: one team in organisation having the authority
- Distributed Incident Response Team — useful for large organisation, e.g., one IRT per division; one IRT per geographic location to enable follow-the-sun model
- Coordinating team — team providing the advice to other teams without having authority
- Staffing models
  - Internal Employees
  - Partially Outsourced
  - Fully Outsourced

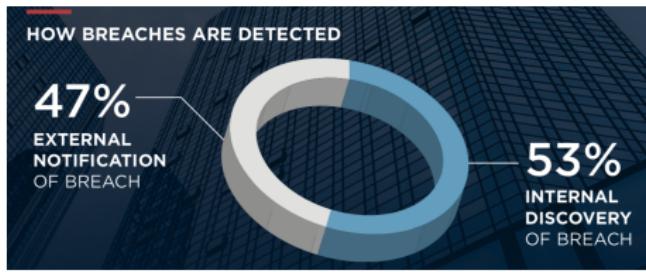
# Detection and Analysis



- Attack Vectors
  - External/removable media
  - Social Engineering: email/phishing, web, ...
  - Brute force attacks
  - Impersonation
  - Improper usage
  - Loss/theft of equipment, ...
- Signs of an incident — precursor, indicator
- Detection?
- Real-time alerts — monitoring
- Users (phishing),  
IT admins (know their systems; observes a suspicious behavior)
- 3<sup>rd</sup> parties — government CERTs; partner organisations observing suspicious traffic outgoing from your network

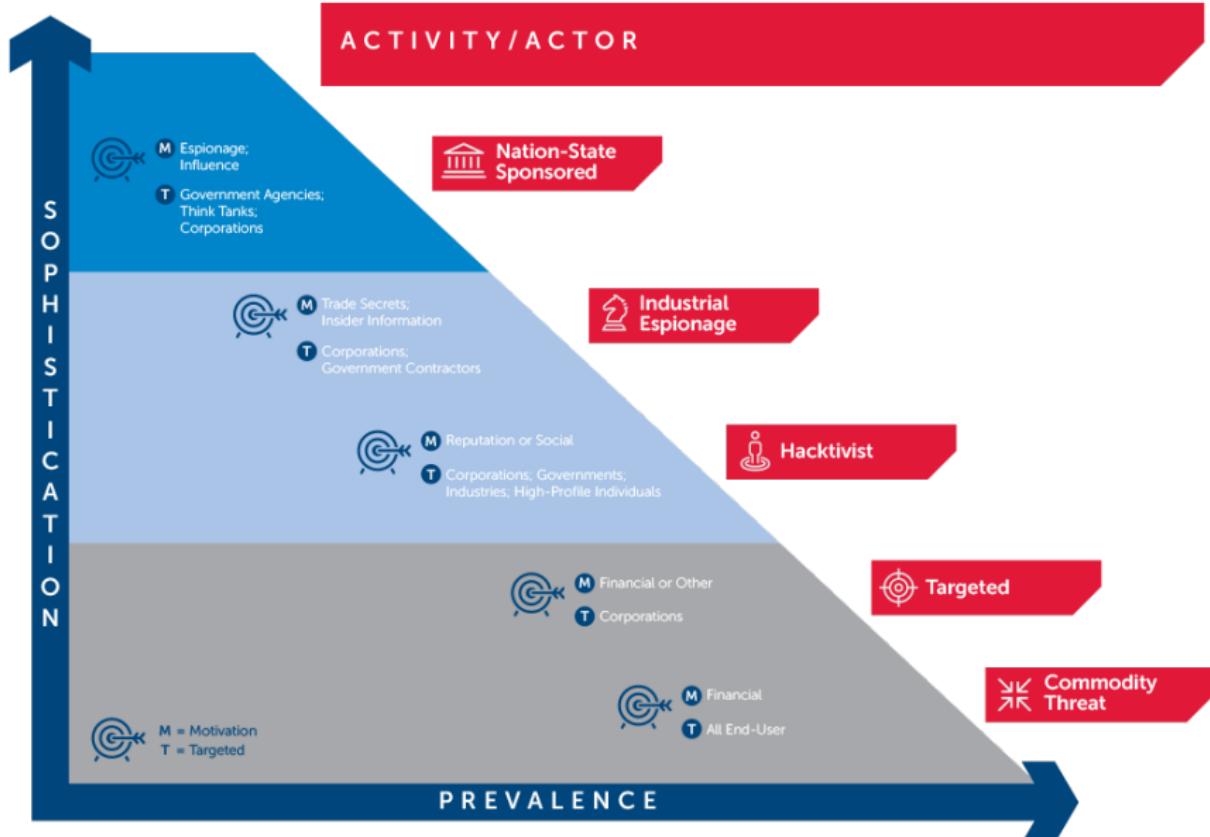
# How we learn about the incident?

source: *FireEye/Mandiant M-trends*



- Dwell time: time from first evidence of compromise that an attacker is present on a victim network before detection
- The global median dwell time is significant: 146 days in 2015, 99 days in 2016, 101 days in 2017
- Actual global dwell times vary significantly, ranging from less than one week to over 2,000 days — depends on complexity of attack/threat actor & maturity of the IR processes and team

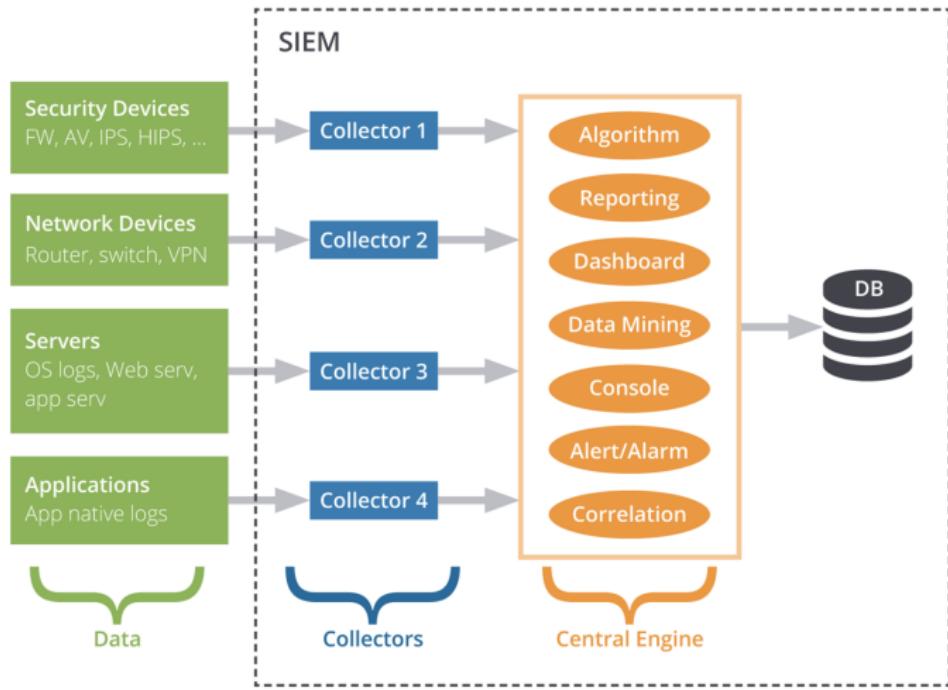
# Threat Actors





- SIEM solutions ingest log data from HW and SW systems, and analyze that data to correlate events and find anomalies or patterns of behavior that may indicate a security breach
- Event and log collection, log management
- Parsing
- Normalisation: reducing the records to just common event attributes (e.g., using data models); known data attributes are fed into a generic template
- Enrichment: adding supplemental information (like geo-location, transaction numbers, mapping known hostnames to IPs, etc.)
- Correlation: looking for patterns of suspicious activities
- *Logging* (continuous activity, no output) vs. *Reporting* (regular automated logs processing with output according to predefined template) vs. *Alerting* (real-time processing of logs)

# Sources of Precursors and Indicators



# Examples of correlations rules



- Detection of specific events, e.g., malware detected event from IDS; suspicious Powershell command observed on the machine; Mimikatz tool detected etc.
- Detection of specific characteristics in network traffic, e.g., large ICMP traffic observed, communication to known malicious domain
- Detection of set of events — multiple failed logons originating from single IP
- Statistical correlations — look for outliers, detect anomalies
- MITRE ATT&CK framework can be used as guidance for creating detections: <https://attack.mitre.org>

# Incident Analysis



- Determining if the event is an incident can be difficult
  - What can help?
    - Profile networks and systems
    - Understand normal Behavior
    - Log retention policy
    - Perform event correlation
    - Keep all host clock synchronized
    - Maintain and use knowledge base, documentation
    - Filter the data
    - Cooperate
- Documentation — once there is suspicion that incident occurred, all the facts regarding the incident should immediately be recorded
- Do not proceed with further steps, until the Analysis is completely finished and the incident properly scoped — missing just one infected machine will allow the re-compromise of the environment again

# Containment, Eradication and Recovery



- *Containment* — actions necessary to prevent further damage (disconnecting system, revoking user access, changing passwords...)
  - First aid: stop the bleeding
  - Even if incident is contained, Eradication & Recovery still needed
- *Eradication*: identify, remove and repair the vulnerability, implement additional security controls ...
- *Recovery*: resuming operations to fully operation status

Technical vs. Managerial vs. Legal

# Evidence Gathering and Handling



- Primary reason — resolve the incident
- Sometimes it may be needed for legal proceedings
  - In such case it is important to clearly document how all evidence, including compromised systems, has been preserved. Evidence should be collected according to procedures that meet all applicable laws and regulations that have been developed from previous discussions with legal staff and appropriate law enforcement agencies so that any evidence can be admissible in court.
- Forensic techniques may be used

# Incident Response vs. Computer Forensics

## Incident Response

- Management and quick containment of the security incident
- Integration into the business processes
- Return back to operations

## Computer Forensics

- Identifying, preserving, analyzing and presenting digital evidence for a legal proceeding
- Detailed and careful handling of digital evidence and analysis

# Creating a Forensic Copy

Susan J. Lincke: *Incident Response*

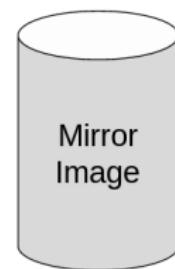
## 2) Accuracy Feature:

Tool is accepted as accurate by the scientific community:



## 4) One-way Copy:

Cannot modify original



## 5) Bit-by-Bit Copy:

Mirror image

## 3) Forensically Sterile:

Wipes existing data;  
Records sterility

1) & 6) Calculate Message Digest:  
Before and after copy

7) Calculate Message Digest  
Validate correctness of copy

# Chain of Custody



- Chain of evidence:
  - whenever evidence is transferred from person to person, chain of custody forms should detail the transfer and include each party's signature. A detailed log should be kept for all evidence, including the following:
    - Identifying information: location, serial number, model number, hostname, ...
    - Name, title, phone, ... of each individual who collected or handled the evidence
    - Time and date of each occurrence of evidence handling
    - Locations where the evidence was stored

# Data Collection



- Offline data collection — machine disconnected, creating exact (bit) copy of hard drive (using write blockers)
- Live data collection — preserves volatile evidence
  - Risks: changes on the system due to collection process (destroying evidence)
- Best practice:
  - Document
  - Use tools with minimum impact on running system
  - Use cryptographic checksums for collected evidence
  - Prefer automation, instead of human interaction with machine
  - Treat all collected data as evidence
  - Consider any data on media connected to the suspect machine as lost, any used credentials as compromised
  - Do not use suspect machine to perform analysis

# Malware Handling



- Safety:
  - Use a virtual environment for triage with isolated network connection
  - Update!
  - Disable convenient features such as drag&drop, clipboard, preview, autoruns
  - Label media containing malware
  - .exe\_, disable execution on the folder
  - Password protected archives (password:infected)
  - Once analysis ready, revert
- *Static analysis* (without execution, disassembling) vs.  
*Dynamic analysis* (run in sandbox & observe the executable's behavior)

# Live Response Collection



- OS and general info (memory, HDD, mounted file systems)
- Running processes
- List of services and programs — autoruns, scheduled tasks
- Local user accounts and group membership; user login history
- Network interface details, routing table, ARP table, DNS cache
- Network connections, including associated processes
- Loaded drivers and modules
- Installed software
- Standard system logs — eventlog; application logs
- MFT (NTFS), inode table, etc.
- Files and other open handles

# Network Evidence



- Event-based alerts
- Header logging
- Full packet logging
- Statistical modeling

# Recovery



- Back to normal operations
- Minimize business processes disruption
- — all remediation steps done

# Lessons Learned



- Do not waste potential of good incident — incidents are drivers for change and improvement
- Lessons learned report & meeting: learn and improve
- Using collected incident data, e.g., feed the risk assessment process (ultimately may lead to implementation of additional controls)
- Review procedures, processes, fix what did not work

# Summary: Incident Response Checklist

	Action	Completed
<b>Detection and Analysis</b>		
1.	Determine whether an incident has occurred	
1.1	Analyze the precursors and indicators	
1.2	Look for correlating information	
1.3	Perform research (e.g., search engines, knowledge base)	
1.4	As soon as the handler believes an incident has occurred, begin documenting the investigation and gathering evidence	
2.	Prioritize handling the incident based on the relevant factors (functional impact, information impact, recoverability effort, etc.)	
3.	Report the incident to the appropriate internal personnel and external organizations	
<b>Containment, Eradication, and Recovery</b>		
4.	Acquire, preserve, secure, and document evidence	
5.	Contain the incident	
6.	Eradicate the incident	
6.1	Identify and mitigate all vulnerabilities that were exploited	
6.2	Remove malware, inappropriate materials, and other components	
6.3	If more affected hosts are discovered (e.g., new malware infections), repeat the Detection and Analysis steps (1.1, 1.2) to identify all other affected hosts, then contain (5) and eradicate (6) the incident for them	
7.	Recover from the incident	
7.1	Return affected systems to an operationally ready state	
7.2	Confirm that the affected systems are functioning normally	

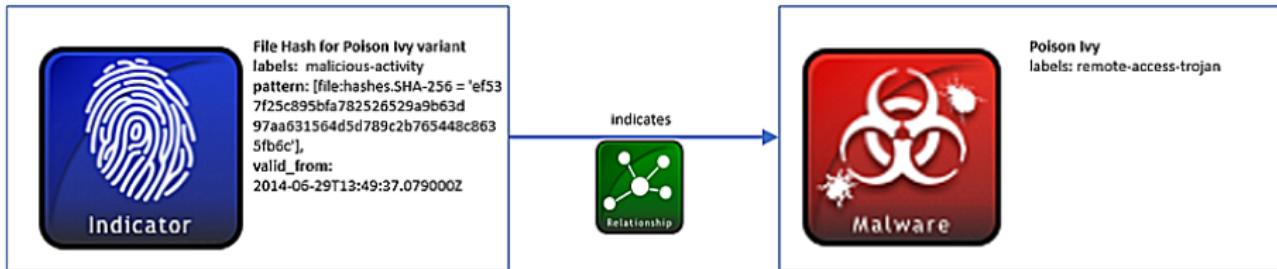
# Bad guys work together, Good guys should too!

## Collaboration & Information Sharing

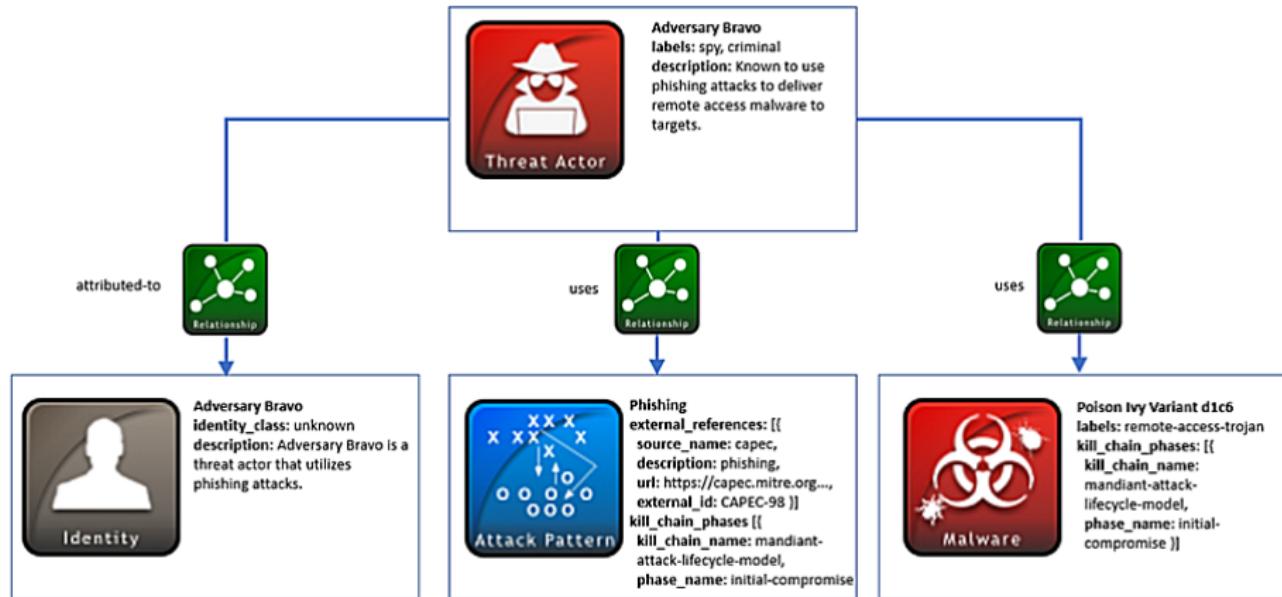
- STIX: A structured language for cyber threat intelligence
- TAXII: A transport mechanism for sharing cyber threat intelligence



# STIX example #1 - Malware Indicator for File Hash



# STIX example #2 - Threat Actor Leveraging Attack Patterns and Malware



## Exercise time: How would you handle... ?

- Data Breach (Password of your users published on Pastebin)
- Emails with links to phishing website
- Hacker Group announced that they will hacked all of YOURDOMAIN on July 1<sup>st</sup> 2014
- Attacker sent you an Email and threatened to launch DDoS attack if you don't pay money
- Receive an email from ShadowServer.ORG about hosts on your network that are running Open Recursive DNS

# Resources

- Susan J. Lincke: Incident Response
- NIST Computer Security Incident Handling Guide
- J. Luttgens et al.: Incident Response and Computer Forensics
- Adli Wahid: Incident Response & Handling
- STIX & TAXII:  
<https://oasis-open.github.io/cti-documentation/>
- <https://idea.cesnet.cz/en/index>

# Questions?

# 11. Penetration Testing

Simona Buchovecká, Tomáš Čejka

Faculty of Information Technology, CTU in Prague

*simona.buchovecka@fit.cvut.cz, tomas.cejka@fit.cvut.cz*

December 7, 2020

- **Vulnerability assessment**

- Using **automated** tools to identify known vulnerabilities in the network
- Only one phase of penetration testing itself! (reconnaissance)
  - Non-exploitable vulnerabilities, False positives
  - Patched != Secure (Design vulnerabilities)

- **Penetration test**

- Identify vulnerabilities and try to exploit them to penetrate a system/network to detect if the vulnerability is genuine
- Usually **manual** effort involved

- **Red team assessment**

- Main goal: to test organization's detection and response mechanisms & processes

# Phases of Penetration Test #1

## 1. Pre-engagement

Scope, testing window, contact information, written approvals, NDAs

## 2. Information Gathering

Passive Reconnaissance (OSINT, sniffing), Active Reconnaissance (social engineering, scanning)

## 3. Threat Modeling

Think like attacker, develop scenario

## 4. Vulnerability Analysis

Vulnerability scanners (Qualys, Nessus, OpenVAS)

# Phases of Penetration Test #2

## 5. Exploitation

Run the exploits, attempt to access client's system

## 6. Post-exploitation

Gather the information about the system, look for interesting files, attempt to elevate privileges . . .

## 7. Reporting

Executive Summary (background, overall posture, risk profile, general findings, recommendations summary), Technical report (all the details)

# Testing Types

- Black box vs. White box vs. Gray box
- Inside attacks vs outside attacks
- Active vs. Passive attacks

# Entry Points

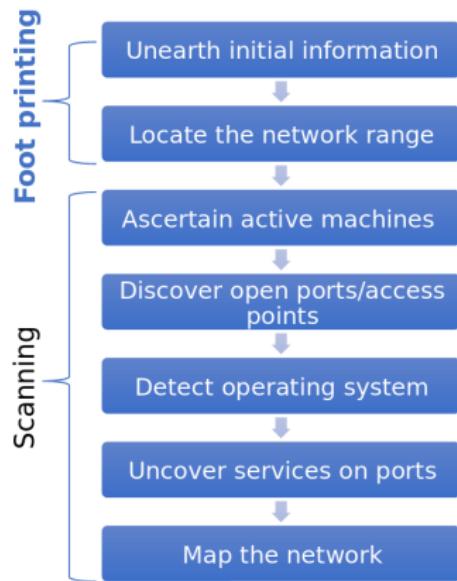
- Remote Network — to simulate an attack launched over the Internet
  - Break a vulnerability in the outside defenses of the network — FW, proxy, router, ...
- Local Network — to simulate someone with physical access to the network and gaining additional unauthorized access, WLANs fall into this category (attacker can stay out of the building)
- Stolen equipment
- Social engineering
- Physical entry

# Reconnaissance

- Term coming from military — active seeking an enemy's intention by collecting and gathering information about an enemy's composition and capabilities via direct observation
- Goals
  - Gather as much information as possible
  - Create list of attackable IP addresses out of this info
- Active vs. Passive

# Information Gathering Methodology #1

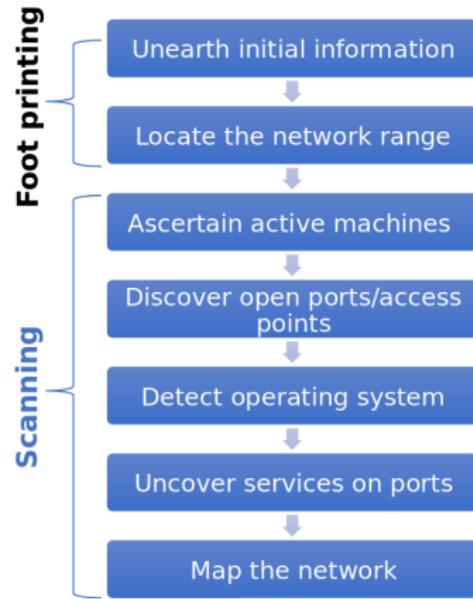
Similar processes/techniques, as the attacker use (and we discussed through the course)



- Web servers
- Whois, URL analytics
- Search engines, webarchives
- Google Earth
- Employee sites, customers
- Financial webs
- Job sites
- Patent/trademarks, Press releases
- Dumpster diving, shoulder surfing, eavesdropping
- Social engineering

# Information Gathering Methodology #2

Similar processes/techniques, as the attacker use (and we discussed through the course)



- Input from foot printing
  - gathered information
  - understanding target
  - list of IPs (that we are authorized to attack)
- Scanning
  - Determine if the system is alive (unreliable, should always continue with next steps)
  - Port scan (to identify the specific ports and services running on the particular host)

# Practice your Google-Fu

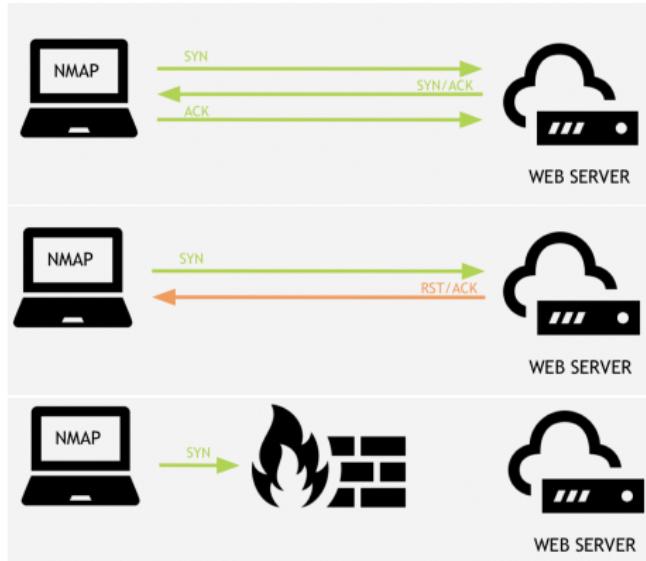
- Useful Google directives
  - site:domain
  - allintitle:, intitle:
  - inurl:
  - filetype:
- Google cache — cache:
- Considered passive only until clicking a link
- Google Hacking for Penetration Testers, Johnny Long's book & DEFCON presentation  
(<https://www.youtube.com/watch?v=fo1BR9itw0Y>)

# Detecting Ping Sweeps & Port Scans

- Almost all IDS/IPS will detect and alert on ping sweep/port scan occurring on network
- Most firewalls and proxy servers block ping responses
- Just because a ping sweep does not return any active hosts on network, it does not mean they are not available

# TCP Scanning

- TCP handshake
- Closed ports
- Filtered ports



# UDP Scanning

- UDP — no confirmation of received data is sent
- UDP packet sent to an open port
  - response: no response
- UDP packet sent to a port that is not open
  - response: ICMP port unreachable
- Absence of response → considering port open
- If port blocked on firewall, false consideration of port being open
- Slow (longer timeout) & unreliable

# Banner Grabbing and OS Fingerprinting

- Banner Grabbing
  - process of opening a connection and reading a banner or response sent by application
  - many applications such as email, FTP, web servers will respond to a telnet connection with the name and version of the software
- OS Fingerprinting
  - Active: sending a data to a system to see how it responds, based on the fact that different vendors implement the TCP stack differently and response will differ based on OS
  - Passive: stealthier, examines traffic on the network. Uses sniffing techniques. Usually undetected, but less accurate.

# Capturing the Traffic

- Insider threat or an attacker who has breached the perimeter simulation
- Capturing the traffic from various systems in the networks can provide us with interesting info (ultimately usernames and passwords)
- Methods to get the traffic that was not intended for us (you know them from previous lectures):
  - ARP Cache poisoning
  - DNS Cache poisoning
  - Encrypted traffic → SSL attacks

# Vulnerability scanning

- Vulnerability: Weakness in the software or system configuration that can be exploited
- Vulnerability scanners (Qualys, Nessus)
  - Check if the remote host is alive
  - Firewall detection
  - TCP/UDP port scanning
  - OS detection
  - TCP/UDP Service Discovery
  - Vulnerability assessment based on the services detected
- cve.mitre.org: Common Vulnerabilities and Exposures system provides a reference-method for publicly known information-security vulnerabilities and exposures
- or <https://www.exploit-db.com/>

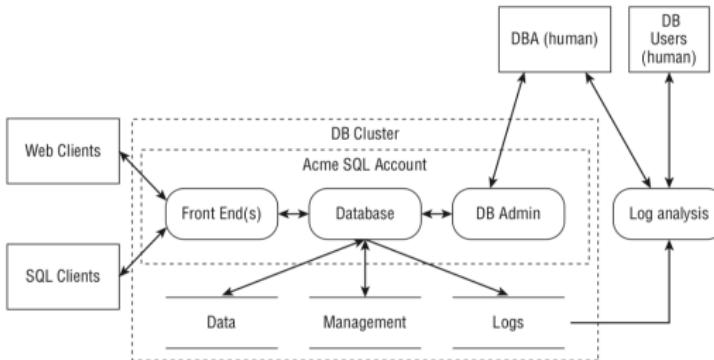
## what is threat modeling?

- **analysis** which exposes *possible threat vectors*, leading to better understanding of a **system**, **asset**, or **attacker** for **defensive** purposes
- primary used as a tool to develop defensive countermeasures
- currently focuses on analysis of system, asset or attacker
- “understand the attack” > “design a compensating defense”
- “how will this be attacked?” “where should we fortify defenses?”

**src:** Offensive Threat Modelling for attackers: [https://media.blackhat.com/bh-eu-12/Los/bh-eu-12-Los-Offensive\\_Threat\\_Modeling-Slides.pdf](https://media.blackhat.com/bh-eu-12/Los/bh-eu-12-Los-Offensive_Threat_Modeling-Slides.pdf)

# Trust Boundaries

## Data Flow Diagram (Example)



Key:



src: <https://users.encs.concordia.ca/~clark/courses/1601-6150/scribe/L04c.pdf>

# Threat Modeling with STRIDE

Threat	Property Violated	Definition	Example
Spoofing	Authentication	Impersonating something or someone else.	Pretending to be any of Bill Gates, Paypal.com or ntdll.dll
Tampering	Integrity	Modifying data or code	Modifying a DLL on disk or DVD, or a packet as it traverses the network
Repudiation	Non-repudiation	Claiming to have not performed an action.	"I didn't send that email," "I didn't modify that file," "I <i>certainly</i> didn't visit that web site, dear!"
Information Disclosure	Confidentiality	Exposing information to someone not authorized to see it	Allowing someone to read the Windows source code; publishing a list of customers to a web site.
Denial of Service	Availability	Deny or degrade service to users	Crashing Windows or a web site, sending a packet and absorbing seconds of CPU time, or routing packets into a black hole.
Elevation of Privilege	Authorization	Gain capabilities without proper authorization	Allowing a remote internet user to run commands is the classic example, but going from a limited user to admin is also EoP.

src: <https://users.encs.concordia.ca/~clark/courses/1601-6150/scribe/L04c.pdf>

## Spoofing On the Local Machine

Threat Example	What the Attacker Does	Notes/Examples
Spoofing a process	Creates a file before the real process	Then your process relies on it
	Abuses names	Create a version of “sudo” and alter PATH
Spoofing a filename	Creates a file in the local directory	Library, executable or config file
	Creates a link, changes it	Also called ‘race condition’ or TOCTOU
	Creates many files in a target directory	Code can easily create all possible /tmp/foo.random

src: <https://users.encs.concordia.ca/~clark/courses/1601-6150/scribe/L04c.pdf>

# Spoofing Over a Network

Threat Example	What the Attacker Does	Notes/Examples
Spoofing a machine	ARP spoofing	
	IP spoofing	
	DNS spoofing	
	DNS compromise	Can be at the TLD, registrar or DNS server
	IP redirection	
Spoofing a person	Take over account	"Stranded in London"
	Set the display name	
Spoofing a role	Declares themselves to be that role	Sometimes opening a special account, setting up a domain/website, other "verifiers"

src: <https://users.encs.concordia.ca/~clark/courses/1601-6150/scribe/L04c.pdf>

# Information Disclosure (Processes)

Threat Example	What the Attacker Does	Notes/Examples
Extracts user data	Exploits bugs like SQL injection to read db tables	Can find this by looking to data stores, but here the issue is the process returning data it shouldn't
	Reads error messages	
Extracts machine secrets	Reads error messages	Cannot connect to database 'foo' as user 'sql' with password '&IO*(^&'
	Exploits bugs	"Heartbleed"

src: <https://users.encs.concordia.ca/~clark/courses/1601-6150/scribe/L04c.pdf>

# Information Disclosure (Data Stores)

Sub-category	What the Attacker Does
Permissions	Take advantage of missing or inappropriate ACLs
	Take advantage of bad database permissions
	File files protected by obscurity
Security	Find crypto keys on disk or in memory
	Get data from logs/temp files
	Get data from swap files
	See interesting information in filenames/directory names
	See data traversing a network
Misc	Obtain device, boot in new OS

src: <https://users.encs.concordia.ca/~clark/courses/1601-6150/scribe/L04c.pdf>

## Information Disclosure (Data Flow)

Sub-category	What the Attacker Does
Network	Read data on a network
	Redirects traffics to enable reading data on the network
Metadata	Learns secrets by analyzing traffic
	Learns who talks to whom by watching the DNS
	Learns who talks to whom by analyzing social network information

**src:** <https://users.encs.concordia.ca/~clark/courses/1601-6150/scribe/L04c.pdf>

# STRIDE #6

## Example Threat Tracking Tables

Diagram Element	Threat Type	Threat	Bug ID
Data flow #4, web server to business logic	Tampering	Add orders without payment checks	4553 "Need integrity controls on channel"
		Payment instruments sent in clear	4554 "need crypto" #PCI

Threat Type	Diagram Element(s)	Threat	Bug ID
Tampering	Web browser	Attacker modifies our JavaScript order checking	4556 "Add order-checking logic to server"
		Failure to authenticate	4557 "Add enforce HTTPS everywhere"

src: <https://users.encs.concordia.ca/~clark/courses/1601-6150/scribe/L04c.pdf>

## how offensive threat modeling differs

- turns focus on the *defenders*
- attempts to understand **defenses**, or **defenders**
- provides analysis of the *weaknesses*
- seeks to develop an **offensive** strategy based on analysis
- primarily useful for stealth-mode attackers
- useful for penetration testing, assessments

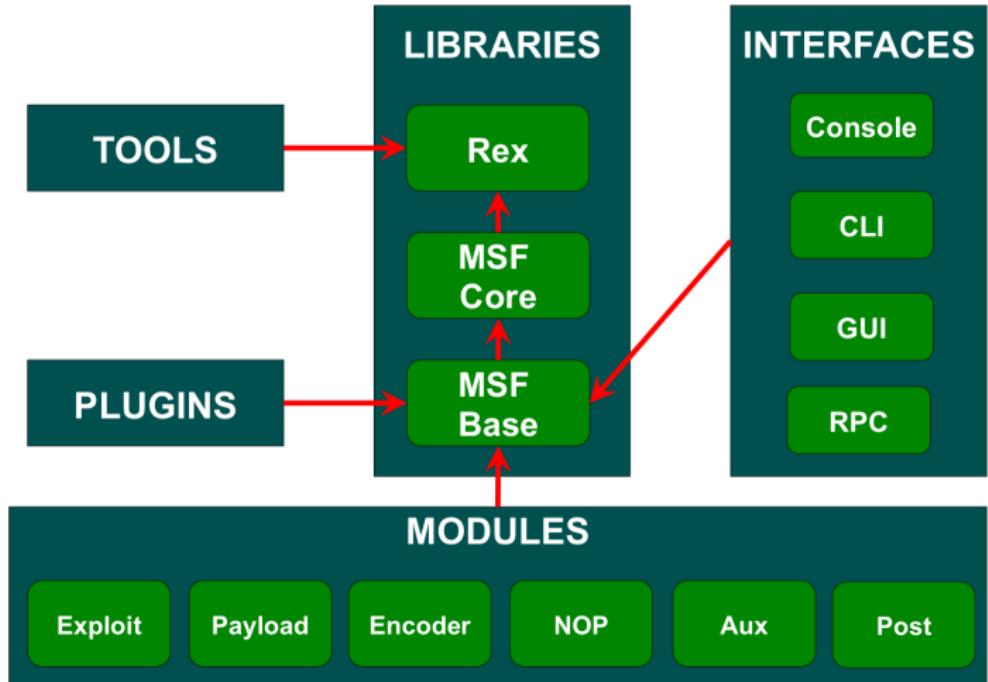
*yes ... this is how an APT will attack you*

**src:** Offensive Threat Modelling for attackers: [https://media.blackhat.com/bh-eu-12/Los-bh-eu-12-Los-Offensive\\_Threat\\_Modeling-Slides.pdf](https://media.blackhat.com/bh-eu-12/Los-bh-eu-12-Los-Offensive_Threat_Modeling-Slides.pdf)

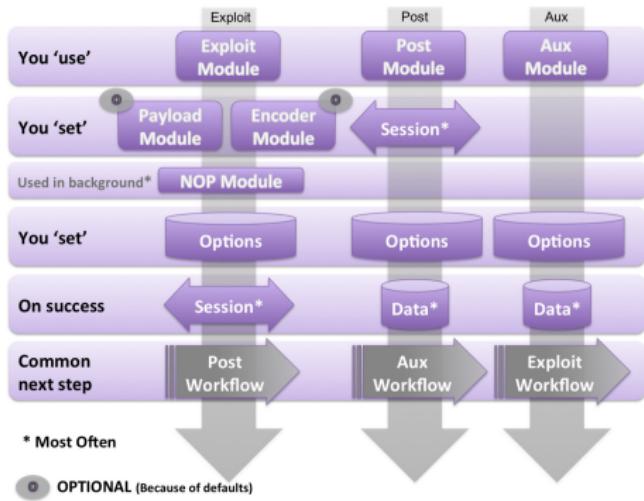
# Exploitation

- The exploitation phase of a penetration test focuses solely on establishing access to a system or resource by bypassing security restrictions
- Countermeasures: Host Based Intrusion Prevention System, Security Guard, Web Application Firewall, or other preventative methods.

# Metasploit Framework



# Metasploit Modules



## • Payloads

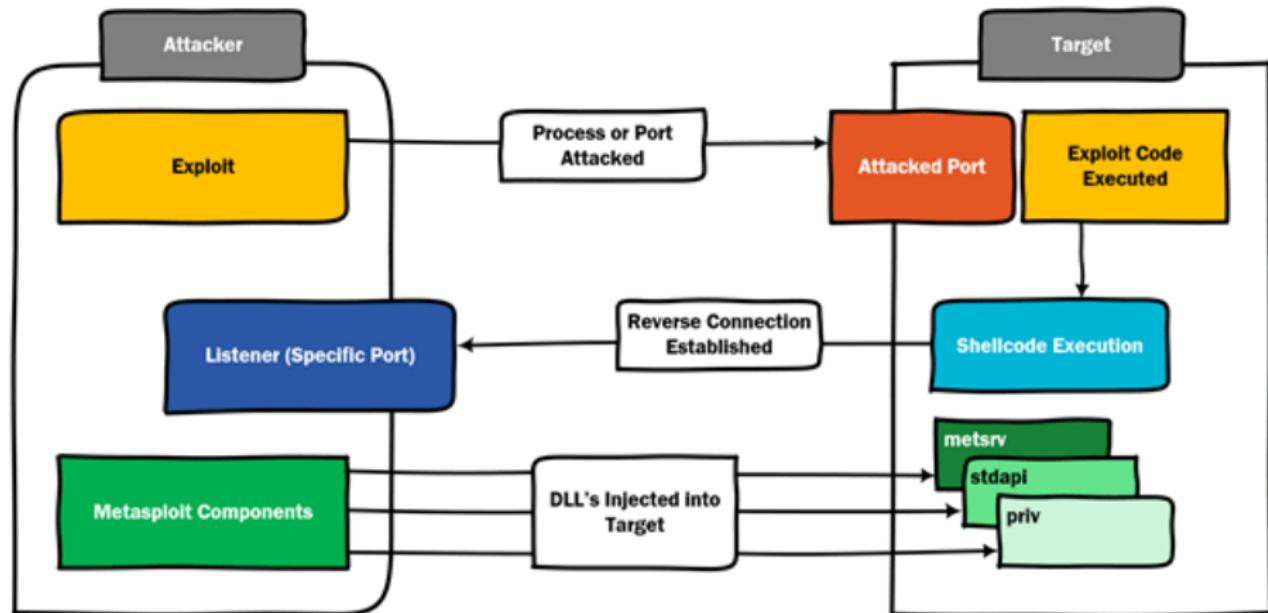
- singles
- stagers
- stages

## • Encoders

- transforming your shellcode into pure alphanumeric, getting rid of bad characters or encoding it for 64 bit target

<https://www.offensive-security.com/metasploit-unleashed/payloads/>

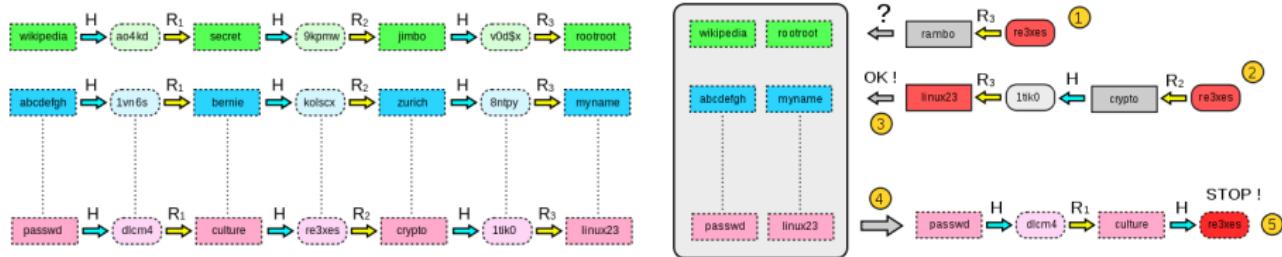
# How it Works



# Password Attacks

- Online vs Offline attacks
- Guessing
- Dictionary attacks
- Brute-force attacks
- Rainbow tables

# Rainbow Tables



[https://en.wikipedia.org/wiki/Rainbow\\_table#/media/File:Rainbow\\_table1.svg](https://en.wikipedia.org/wiki/Rainbow_table#/media/File:Rainbow_table1.svg)

[https://en.wikipedia.org/wiki/Rainbow\\_table#/media/File:Rainbow\\_table2.svg](https://en.wikipedia.org/wiki/Rainbow_table#/media/File:Rainbow_table2.svg)

# Fuzzing

- Automated software testing technique that involves providing invalid, unexpected, or random data as inputs to a computer program.
- The program is then monitored for exceptions such as crashes, or failing built-in code assertions or for finding potential memory leaks.

# Post Exploitation

- Determine the value of the machine compromised and to maintain control of the machine for later use
- Value of the machine is determined by the sensitivity of the data stored on it and the machines usefulness in further compromising the network

# Persistence

- Installation of backdoor that requires authentication.
- Installation and/or modification of services to connect back to system. User and complex password should be used as a minimum; use of certificates or cryptographic keys is preferred where possible. (SSH, ncat, RDP). Reverse connections limited to a single IP may be used.
- Creation of alternate accounts with complex passwords.
- When possible backdoor must survive reboots.

# Local Information Gathering

- Infrastructure analysis – network configuration & services, installed services
- Sensitive data
  - key logging, screen capturing
  - network traffic capture
- User information
- System Configuration
- Files

# Lateral Movement

- Application Deployment Software
- Logon Scripts
- Pass the Hash
- Pass the ticket
- Remote Desktop
- Remote Copy File
- Remote services
- Windows Admin Shares
- PSEXEC

# Cleanup

- Remove all executables, scripts and temporary file from a compromised system. If possible use secure delete method for removing the files and folders.
- Return to original values system settings and application configuration parameters if they where modified during the assessment.
- Remove all backdoors and/or rootkits installed.
- Remove any user accounts created for connecting back to compromise systems.

# References

- <http://www.pentest-standard.org/>
- Kimberly Graves: Certified Ethical Hacker Study Guide
- Georgia Weidman: Penetration Testing

# Questions?