Analyzing Internet Attacks with Honeypots

Ioannis Koniaris – *ikoniaris@gmail.com*

Workshop outline
About me

Workshop outline

- Cyber threats and countermeasures
  - Information and systems security
  - Human threat agents and malicious software
  - Firewalls, Intrusion detection systems (IDS)
- Honeypots
  - Introduction and definitions
  - History / timeline
  - Emulation of OSes and services
  - Honeypot classifications
    - Based on purpose, based on interaction level
  - Network placement and operation

Workshop outline

- Honeypots (cont.)
  - Other honeypot-related technologies
    - Client honeypots
  - Value of honeypots for information and network security
    - Advantages / Disadvantages
  - Legal issues concerning honeypot operations
- Kippo SSH honeypot
  - Introduction
  - Hands-on lab!!! ☺
    - Setup and configuration
    - Attack analysis and visualization

Workshop outline

- Dionaea malware honeypot
  - Introduction
  - Hands-on lab!!! ☺
    - Setup and configuration
    - Attack analysis and visualization
- HoneyDrive distro hands-on lab
  - Introduction / guided tour
- QnA – proposals for new research projects!

Whoami

- DevOps, Security, Software Engineer (start-up anyone?)
- Academic research on honeypots/nets (AUTH: Aristotle Univeristy of Thessaloniki, Greece)

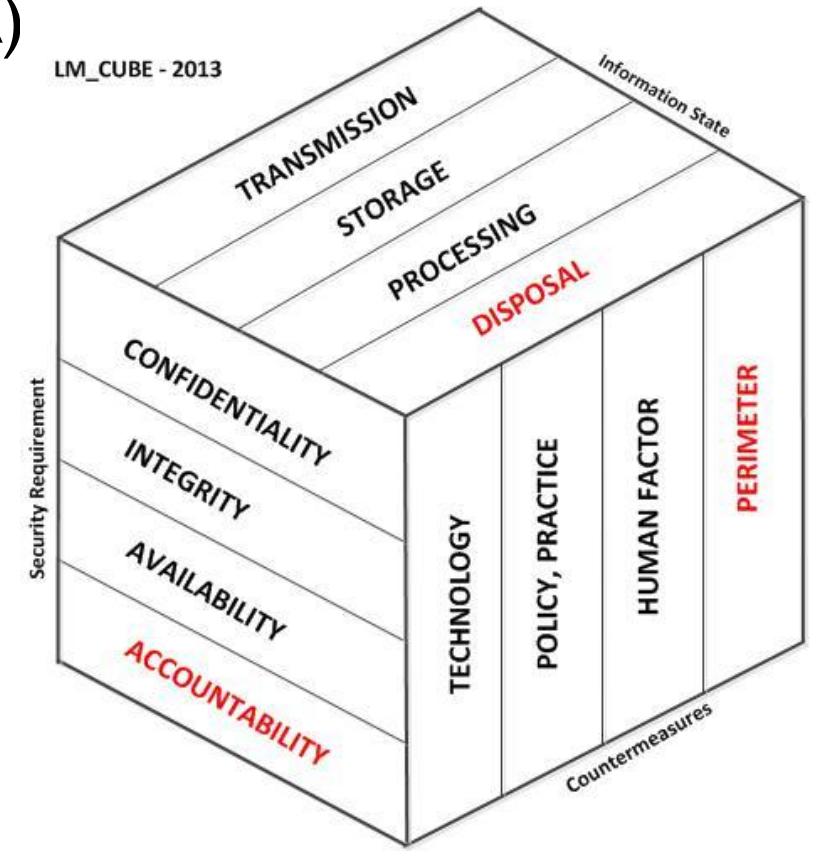
- BruteForce Lab – http://bruteforce.gr
- Twitter: @ikoniaris
- Email: ikoniaris@gmail.com

- Interests: honeypots, honeynets, botnet tracking, malware analysis, security visualization

Information security
Human & machine threats

Information security

- Protection of information – an asset with special meaning
- Information security concepts and goals: confidentiality, integrity, availability (CIA)
- McCumber Cube (1991)
- LM Cube (2013)



Scope of security operations

- Prevention
  - OS, software patching
  - Perimeter setup, firewalls, etc
  - Security policies
- Detection
  - Intrusion detection systems (IDS)
  - Security monitoring
  - Honeypots!
- Response
  - Incident handling
  - Forensic examination

Human threat agents

- Low level attackers
  - "Script kiddies"
  - Quick and easy break-in
  - No real knowledge, just using tools and exploits
  - The majority of attacks comes from script kiddies
  - They seem harmless but they have frequent successes!
  - Social acceptance, bragging rights, curiosity, political activism
- Medium level attackers
  - Knowledgeable about security topics in general
  - Understanding of the nature of vulnerabilities
  - Can configure tools and hide tracks during exploitation
  - Financial motivation (phishing, spamming, etc), political activism

Human threat agents

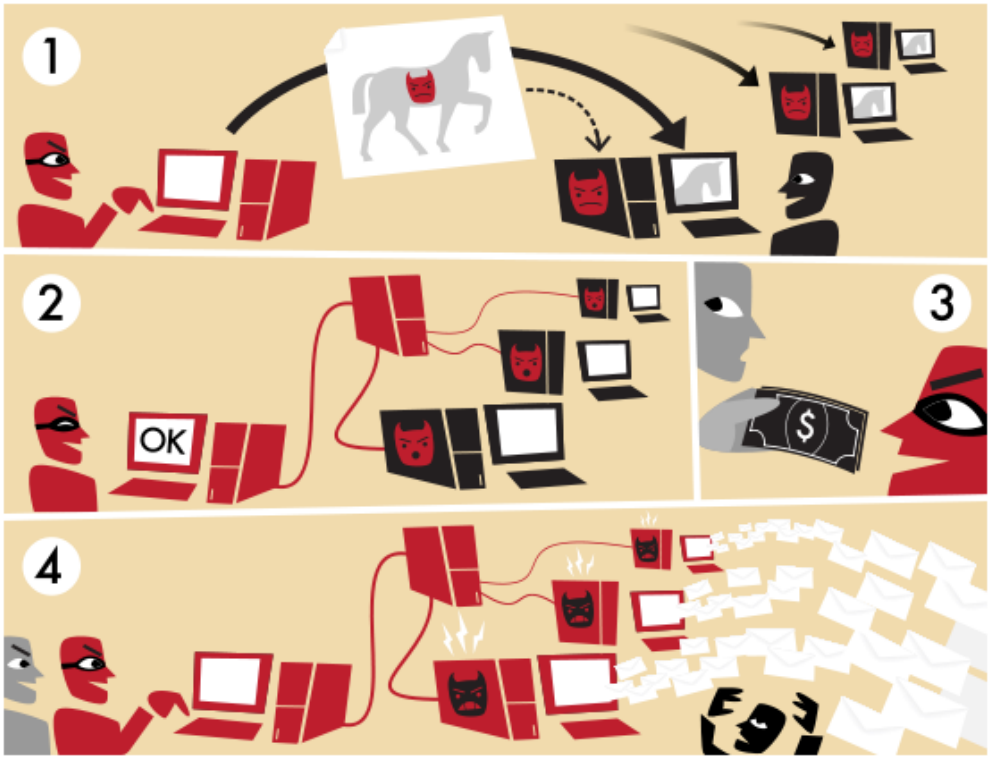
- High level attackers
  - Advanced knowledge of security field and specific topics in particular
  - Can find new and unknown vulnerabilities and write exploits
  - Can hide and cover their tracks with advanced techniques
  - Pick (high profile) targets slowly and methodically
  - Hired guns, financial motivations, political activism

Malware (malicious software)

- Worms
  - Self-propagating malware
  - 3 phases: infect, attack, spreading
  - Attack mostly common security vulnerabilities in a continuous fashion (secondary: mail, P2P, IRC, etc)
- Viruses (it's 2013, I know ☺)
  - Don't auto-propagate
  - Infect other programs, spread mostly through mail
- Trojan horses
  - Keylogging, backdoors
  - Usually camouflage themselves as legitimate software

Malware (malicious software)

- Rootkits
  - Stealthy backdoor access
  - High level of hiding inside the system
- Bots
  - Autonomous programs
  - Botnets
  - Zombies, C&C servers, bot herder/master
  - Spamming, phishing, illegal financial gains



Offensive countermeasures
Common weaknesses

Firewalls

- A device on the perimeter or inside a network, allowing or disallowing packets based on specific criteria
- Stateless filtering
  - Packet headers
  - Quick, easy, low security
- Stateful inspection
  - Creation of a table with client state/connections
  - Allow connections from external networks if initiated from the protected network (e.g. websites)
  - Medium performance, medium security
- Deep Packet Inspection
  - Combination of stateful filtering and IDS
  - Examining content on a higher layer than that they need to

Firewall weaknesses

- A firewall cannot protect the network from attacks that can bypass it
- A firewall cannot protect the network from inside threats and internal attacks
- A firewall cannot protect host machines from getting infected by malware
- Vulnerabilities in the firewall appliances themselves

Intrusion detection systems (IDS)

- Intrusion detection: a process that can identify anomalous, non-compatible, erroneous or generally suspicious activity
- Types of IDSes
  - Network based (NIDS)
    - Packet checking, content filtering
  - Host based (HIDS)
    - System monitoring for suspicious changes
- Attack detection
  - Signature based
    - Pattern matching, someone must create these patterns first!
  - Anomaly based

IDS weaknesses

- Networks mostly use switches, so NIDSes need to be placed in front of them but this cannot secure the network from the inside
- Throughput and "power" of the IDS can be limited
- IDSes produce data overload: many false positives!
- IDSes cannot detect or identify new attacks and exploits
- IDSes need expensive and high-tech hardware in order to perform efficiently
- IDSes normally cannot process encrypted data (SSL, IPSec, etc)
- → Honeypots can give solutions to both firewalls and IDS weaknesses!

Honeypots: definitions & intro
History and current state

Honeypots

- Definition:
  - "An information system resource whose value lies in unauthorized or illicit use of that resource" (Lance Spitzner)
- It's a system with no production value
  - There is no reason for a legitimate user to use it or interact with it
  - Any communication attempt with the system is automatically considered malicious
  - A honeypot that tries to connect to another resource is probably compromised
- They are both deceit tools and traps
  - Attackers waste time while their actions are monitored closely

Honeypots

- They cannot prevent attacks against the network by themselves
- But, they can help in the detection phase of an attack and identify the target and methods of exploiting
- They can be used in conjunction with firewalls and IDSes and in fact "complete" their role by substituting for their weaknesses

OS and service emulation

- Honeypot OS emulation is done using the so-called "fingerprints"
  - A fingerprint comes from the IP stack of an OS, as 8 parameters of the TCP/IP protocol are not stable – 67 bit signature
  - Different OSes and different versions of the same OS have distinct fingerprints
  - It's the same way that various tools identify the remote OS, e.g. nmap, p0f, etc
- Service emulation is done using scripts with identical behavior and output as the real services

The history of honeypots

- Early 90's, publications: "The Cuckoo's Egg" by Cliff Stoll, "An evening with Berferd" by Bill Cheswick
- 1997: Deception Toolkit (DTK) by Fred Cohen
  - The "grandparent" of today's low interaction honeypots
  - Perl scripts, emulating various vulnerable network services
- 1998: CyberCop Sting, the first commercial honeypot
  - Ran on Windows NT
  - It could emulate a whole network of computers using different fake IP stacks, not just one system
- 1998: NetFacade, another commercial honeypot
  - Limited success, but inspired the creation of Snort IDS

The history of honeypots

- 1999: Formation of the non-profit "Honeynet Project"
- 1999: ManTrap (rebranded as Decoy Server by Symantec)
  - Up to 4 machines, fake network traffic between them
- 2002: Tiny Honeypot by George Bakos
  - Written in Perl, listens for connections on every unused port
- 2002: Honeynet Research Alliance by Honeynet Project
  - An effort to deploy many honeypots in various address spaces around the world and share results
  - 2003: Snort_inline, Sebek, virtual honeypots

Current state of honeypot software

Honeypot	Type	OS	Language	GUI	License
Honeyd	Generic	LINUX	C	N	GNU
Nepenthes	Malware	LINUX	C	N	GNU
Dionaea	Malware	LINUX	PYTHON	N	GNU
Honeytrap	Generic	LINUX	C	N	GNU
LaBrea	Generic	LINUX	C	N	GNU
Tiny HP	Generic	LINUX	PERL	N	GNU
HoneyBot	Malware	WINDOWS	-	Y	CLOSED
Google Hack HP	WEB	-	PHP	Y	GNU
Multipot	Malware	WINDOWS	VB 6	Y	GNU
Glastopf	WEB	-	PYTHON	Y	GNU
Kojoney	SSH	LINUX	PYTHON	N	GNU
Kippo	SSH	LINUX	PYTHON	N	BSD
Amun	Malware	LINUX	PYTHON	N	GNU
Omnirova	Malware	WINDOWS	Borland Delphi	Y	GNU
BillyGoat	Malware	-	?	?	CLOSED
Artemisa	VOIP	-	PYTHON	N	GNU
GHOST	USB	WINDOWS	C	Y	GNU

Honeypot classifications

Honeypot classifications

- Honeypots can be divided into categories based on two criteria:
  - A) The purpose of honeypot deployment
  - B) The level of allowed interaction with the honeypot

- Honeypot categories based on purpose:
  - Production honeypots
  - Research honeypots

Honeypot classifications

- Production honeypots
  - Placed along the real systems of a business, acting as decoys
  - Ideally they are mirrors of real servers where attackers will waste their time and effort while we are gathering intelligence on their methods and attack vectors
  - As said before, they cannot prevent attacks all by themselves!
- Research honeypots
  - Their main goal is to monitor attack activities and capture malicious connections, network traffic and files
  - Their data are crucial to enhance the understanding of threat agents and their ways of operation
  - Usually deployed by researchers, universities, non-profits (e.g. Honeynet Project), military/goverment agencies
- In our workshop we focus on research honeypots
  - You are free to implement production honeypots at work ☺

Honeypot classifications

- Honeypot categories based on the level of allowed interaction:
  - Low-interaction honeypots
  - Medium-interaction honeypots
  - High-interaction honeypots

- Low-interaction honeypots
  - As the name implies, they offer little to no interaction between the server and the attacker
  - It's not a real system, but software emulating one or more network services
  - Low added risk to the network, but it only logs connection attempts: date and time, source IP and port, destination port

Honeypot classifications

- Medium-interaction honeypots
  - Offer greater interaction between the system and the attacker
  - The emulated network services respond to the attacker and allow access to fake resources (e.g. a fake FTP server)
  - Can be used to catch malware as well by emulating specific vulnerabilities in a service
  - Medium added risk but generally good results and data!
- High-interaction honeypots
  - A real vulnerable OS given to attackers as sacrificial lamb
  - Intruders will have real access and control of the system
  - High risk and high reward! Greatest level of data capture, BUT they must be isolated and monitored at all times! (pivoting)

Honeypot classifications

	Low-interaction	Medium-interaction	High-interaction
Interaction level	Low: no access	Medium: controlled access	High: full access
Real OS	No	No	Yes
Risk level	Low risk	Medium risk	High risk
Data collection	Limited: only connection attempts	Varied: depending on intruder skills	Extensive: all available data
Setup & config	Easy	Easy/Medium	Hard
Maintenance	Easy	Medium	Hard

Placement and operation

Network placement and operation

- Mainly 3 common honeypot placement spots:
  - A) Externally, in front of the firewall, facing the Internet
  - B) Internally, behind the firewall
  - C) Demilitarized Zone (DMZ)
- External placement
  - Used when trying to immediately make them available to attackers for intrusion and takeover
  - Most suitable for sole research honeypots
  - Honeypots and other network hosts share the same subnet
  - One or more public IPs are needed
    - If only one is available, it's assigned to the honeypot and a monitoring station takes a private address

Network placement and operation

- Internal placement
  - Most suitable to detect attackers (human or software) that have breached the perimeter
  - Effective early warning system
  - High added risk to the network if using a high-interaction honeypot and it gets taken over – egress firewall needed
  - Ingress rules needed as well, mainly port forwarding: all ports that are not being used can be forwarded to the honeypot
- DMZ placement
  - Best choice for a business/organization
  - Honeypots and other DMZ hosts share the same subnet
  - Can be setup as mirrors of real systems in order to catch early attacks against them DMZ

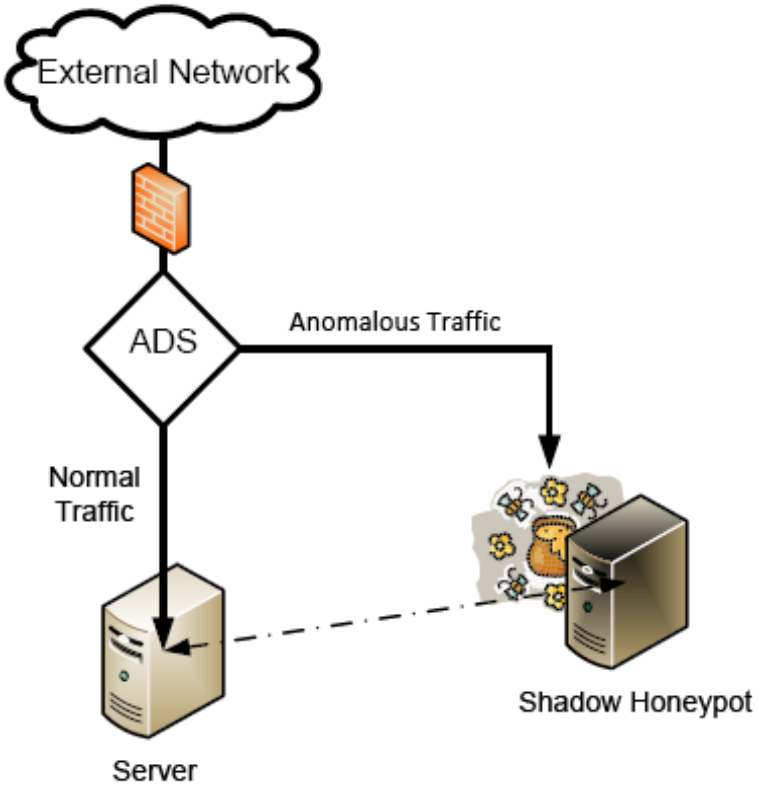
Other related technologies

Other honeypot-related technologies

- Honeytokens
  - An object with no production value placed in a system as an intrusion detection mechanism
  - Various small electronic baits that no legitimate user should access – e.g. fake admin account user/pass combination
  - If a honeytoken is found in the application's logs, the system has been compromised
- Honeypages
  - Fake web pages inside a real web app, with no production value
  - There is no direct link to them, every request is considered malicious
  - A request can come from automated scanning, robots.txt analysis, etc – honeypages log every info they can get

Other honeypot-related technologies

- Shadow honeypots
  - Combination of honeypot and ADS (Anomaly Detection System) – an alternative solution as a rule-based IDS
  - "Suspicious" traffic is forwarded to a honeypot which is a mirror of the real application
  - If an attack occurs the honeypot resets its state and no harm is done, if the traffic is OK it is forwarded to the real server



Client honeypots

- Targeting servers is so 2008!! ☺☺
- Attackers nowadays target client programs (browsers, media players, file viewers etc)
- A client honeypot doesn't wait passively for attacks to come to it, but actively tries to find malicious websites serving exploits targeting client applications
- They usually use HTTP, and emulate various web technologies like JavaScript, Active-X, etc like a browser
- Three part model: queuer, the client, analysis engine
- Like traditional honeypots they are also classified into low and high interaction

The value of honeypots
Advantages & disadvantages

Value of honeypots for network security

- Honeypots present a unique concept and very valuable for information and network security
- They give almost no false positives
  - One the of the biggest problems for IDS analysts is the "noise" generated by their systems
  - Honeypots have no production value and thus any interaction with it can be automatically considered malicious and candidate for further analysis
- They help us detect malicious actions early on
  - Continuing from the previous point, they can detect real attacks fast
  - Sysadmins can use them to quickly classify the nature and severity of attacks

Value of honeypots for network security

- New threat identification
  - Every connection destined to a honeypot is considered malicious and the actor behind it a threat
  - New and unknown attacks can be logged and identified as malicious as fast as common attacks
  - Tools like Honeycomb can create IDS rules from these in order to increase defense levels in a larger scale
  - Also, any tools downloaded or content generated (e.g. IRC logs) by attackers are saved for further analysis
- Add an extra layer of protection (Defense-in-Depth)
  - For example when they are placed internally in order to catch inside threats or warn sysadmins for malicious software

Other advantages of honeypots

- 1. Simplicity of their idea: well known technology, not very hard for a sysadmin to implement them
- 2. Can be used as deceit systems: they can make attackers waste time and effort on fake systems
- 3. They provide a small amount of captured data of high value: easy to analyze dataset and extract information
- 4. Can catch early threats and attacks before they can cause harm or damage: e.g. honeypots as mirrors of real production systems can give us early warnings
- 5. Low requirements in terms of hardware: even a Pentium can run a modern honeypot! (yes, I've tried ☺)
- 6. Honeypots can be as effective in crypto environments (where IDSes might have problems) or IPv6 networks

Disadvantages of using honeypots

- 1. No real value if nobody attacks them! ☹
- 2. They have a very limited attack detection radius/scope, as they can catch attacks only against themselves
- 3. Compromised honeypots can be used as platforms to launch further attacks against the network (pivoting)
- 4. Honeypots can sometimes taunt attackers and thus increase the level of overall risk
- 5. Honeypots themselves can contain bugs or vulnerabilities that either make them targets in a traditional sense or make them easily identifiable
- 6. Placing honeypots increases the overall complexity of a network – not good from a security standpoint

Legal issues with honeypots

Legal issues concerning honeypot usage

- Using and operating honeypots present some legal challenges, due to the nature of these systems
- Different legislations across the world, different laws concerning the acquisition and storage of data
- No definite answer can be given, even though all top honeypot researchers agree that we are on the safe side!
- Some of the specific legal issues:
- A) Privacy:
  - Essentially the question: "how much data can an admin gather and store before a privacy problem arises?"
  - Is it legal for an admin to capture data from other company employees? What about external threats in general?

Legal issues concerning honeypot usage

- Privacy (cont.)
  - According to Lance Spitzner (using info from the US legal system; specifically laws governing forensic investigations and obtaining evidence):
  - The people breaking into these systems are NOT AUTHORIZED to use them, and if they place any files on them, they have given up their privacy rights to that data by placing it on the honeypot
  - By using honeypots for communication, they have given up their right to privacy in that communication, as honeypots are not service providers and are not bound by privacy requirements designed for service providers

Legal issues concerning honeypot usage

- B) Entrapment
  - *A person is 'entrapped' when he is induced or persuaded by law enforcement officers or their agents to commit a crime that he had no previous intent to commit*
  - Again, setting up honeypots cannot be considered an entrapment activity because:
  - Honeypots do not induce or persuade anyone, neither promote malicious activity by themselves only
  - Attackers find and attack honeypots based on their own initiative
  - Most sysadmins are not law enforcement agents and they are not using honeypots to collect evidence and prosecute, but instead as means to detect and possibly learn about attacks

Legal issues concerning honeypot usage

- C) Liability
  - Hypothetical scenario: a honeypot of company X is compromised by an attacker and is used as the source of attacks against the network of company Y.
  - Who's to blame??? 😐
  - No definite answer in this case, BUT we should also have mitigated this risk in the first place! (firewalls, egress filtering, etc)

Hands-on lab preparation

VM SETUP FOR HANDS-ON LAB

- Pass around the DVD(s)
- Install VirtualBox or download: http://www.virtualbox.org
- Copy the HoneyDrive OVA file to your HDD or download: http://bruteforce.gr/honeydrive
- Double-click on it to import it in VirtualBox (~15min)
- Copy the "dataset" folder to your HDD or download: http://bruteforce.gr/brucon-dataset.zip

- Let the game begin!!! ☺

Kippo SSH honeypot
Kippo-Graph

Kippo SSH honeypot

- https://code.google.com/p/kippo/
- Kippo is a medium-interaction honeypot
- Written in Python (Twisted)
- It logs the entire shell session (UML compatible)
- Also saves all the files downloaded by attackers

- Kippo emulates a Debian 5 OS
- You can add/edit/remove files
- You can add fake file content (e.g. /etc/passwd, etc)
- You can add fake command output (e.g. ifconfig, ssh, etc)

Kippo SSH honeypot

- Online guides:
  - http://bruteforce.gr/installing-kippo-ssh-honeypot-on-ubuntu.html
  - http://bruteforce.gr/logging-kippo-events-using-mysql-db.html
- Interesting stuff:
  - dl folder, log/kippo.log, log/tty, utils/playlog, fs.pickle, honeyfs folder, data/userdb.txt, kippo.cfg
- MySQL schema:
  - auth, clients, input, sensors, sessions, ttylog
- Visualization with Kippo-Graph
  - http://bruteforce.gr/kippo-graph
- **DEMO TIME!**

Dionaea malware honeypot DionaeaFR

Dionaea malware honeypot

- http://dionaea.carnivore.it/
- The successor of Nepenthes honeypot
- Written in C/Python
- Emulates protocols, not vulnerabilities per se
- Mail protocol: SMB (CIFS), port 445
- Other protocols: HTTP(S), (T)FTP, MSSQL, MySQL, SIP

- Dionaea uses libemu to detect and analyze shellcodes (profiling – GetPC)
- Shellcodes run inside a libemu VM and API calls get recorded

Dionaea malware honeypot

- Different types of payloads
  - Shells: bind or connectback – Dionaea emulates cmd.exe, parses the input and acts accordingly
  - URLDownloadToFile: uses the API call to download a file through HTTP and executes it locally
  - Exec: some shellcodes use the WinExec API call – Dionaea behaves like in the bind/connectback case
  - Multi-stage payloads: the first stage receives a second payload, shellcode is executed inside the libemu VM

Dionaea malware honeypot

- Malicious file download
  - After analyzing the shellcode Dionaea tries to download the malicious binary from the extracted web address
  - FTP and TFTP downloads are implemented in Python, HTTP is done using libcurl
  - Files are stored locally for further analysis, and Dionaea can also send them to online malware analysis services (VirusTotal, CWSandbox, Anubis, Normal Sandbox)
- Logging
  - Text based logging can be very difficult to parse/analyze, although you can use filters
  - Dionaea uses "incidents" and "ihandlers"

Dionaea malware honeypot

- Logging (cont.)
  - "incidents": Dionaea's internal communication system
  - Every incident has an origin, a path and some properties
  - Every incident is handled by an appropriate incident handler called "ihandler"
  - Dionaea can use a number of different ihandlers, e.g. logsql, p0f, virustotal, etc

- Online guides:
  - http://bruteforce.gr/starting-with-dionaea-malware-honeypot.html
  - http://bruteforce.gr/some-dionaea-statistics.html
  - http://bruteforce.gr/visualizing-dionaeas-results-with-dionaeafr.html

Dionaea malware honeypot

- Interesting stuff:
  - dionaea/bin, dionaea/etc/dionaea, dionaea/var/log, dionaea/var/dionaea, dionaea/var/dionaea/binaries, dionaea/var/dionaea/bistreams
- Configuration
  - logsql, p0f, virustotal (ihandlers)
- Analysis (readlogsqltree, phpliteadmin, dionaea-scripts)
- Visualization (gnuplotsql, DionaeaFR)

- **DEMO TIME!**

Questions & answers
General discussion

Feedback!

- PLEASE, don't forget to send me your feedback and suggestions: ikoniaris@gmail.com

- Some other contact info:
  - BruteForce Lab – http://bruteforce.gr
  - Twitter: @ikoniaris

- And again, some of my interests:
  - honeypots, honeynets, botnet tracking, malware analysis, security visualization

Thanks BruCON, it has been a pleasure!!!