Computer Security

System Software Security

Prof. Jean-Noël Colin
jean-noel.colin@unamur.be
Office #306

University of Namur
Computer Science Faculty

www.unamur.be

UNIVERSITÉ DE NAMUR

Systèmes d'information
Méthodes et technologies

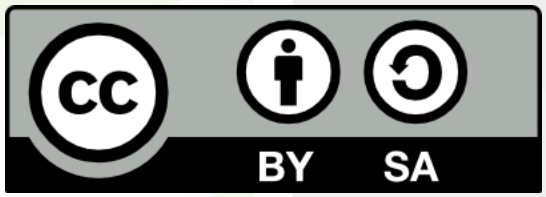
System Software Security

Prof. Jean-Noël Colin
jean-noel.colin@unamur.be
Office #306

University of Namur
Computer Science Faculty

www.unamur.be

UNIVERSITÉ DE NAMUR



Agenda

- Introduction
- Threats and counter-measures
- OS management
- Backup
- Monitoring

Operating System

- Examples
  - Microsoft family: Windows 98, 2000, XP, Vista, Windows 7, Windows 10, Server 2012, Server 2016, Windows Phone…
  - Unix family: Solaris, HP-UX, Linux, FreeBSD, OS X…
  - Mobile OS: iOS, Android (Linux based), Symbian…
  - Contiki OS, RTOS…
- Missions
  - Manage resources (cpu, memory, peripherals)
    - allocate
    - share (time/space multiplexing)
    - control access
  - Abstraction layer
    - simple and stable interface
    - hardware independence
- Last line of defense

Security objectives and controls

- Confidentiality – avoid data leak through
  - access control through proper permission management
  - cryptographic controls
  - auditing and reporting
- Integrity – avoid data corruption through
  - access control through proper permission management
  - cryptographic controls
    - file integrity
    - code signature (Solaris's Signed ELF Objects, MS's Sign Tool)
- Availability
- Traceability

Essentially, a matter of access control

Threats and counter-measures

Various threats

- Impacting access
  - Backdoor
    - Maintain a hidden entry point
  - Login spoofing
  - Privilege escalation
- Logic bomb

Dos – Denial of Service

- DoS (1974) or DDoS (1999)
  - Connectionless attacks
    - [UDP Flood] flood victim with UDP datagrams on random ports; IP spoofing to avoid detection
    - [ICMP Flood] idem with ICMP requests
  - Connection-oriented attacks
    - [TCP SYN Flood] initiate TCP handshake without completing it; causes resource depletion on target. IP spoofing to avoid detection.
    - [TCP RST] send RST to close an open connection (requires knowledge of the sequence number)
    - [Sockstress] send 'window size=0', which blocks further transmissions until 'window size' is changed back
  - HTTP flood, DNS flood, vulnerability exploit (regex DoS attack...)

DoS – Denial of Service

- Reflection attack: use a third-party system to mask the identity of the attacker
- Amplification attack: volume of data received by victim is higher than volume sent by attacker
  - [DNS Amplification attack] spoof victim's IP and send DNS requests to multiple DNS resolvers that send large responses back
  - [Smurf Attack] spoof victim's IP and send ICMP requests to broadcast address

Malware

Security

NotPetya ransomware attack cost us $300m – shipping giant Maersk

IT crippled so badly firm relied on WhatsApp

By Iain Thomson in San Francisco 16 Aug 2017 at 22:15 29 SHARE ▼

ship

The world's largest container shipping biz has revealed the losses it suffered after getting hit by the NotPetya ransomware outbreak, and the results aren't pretty.

The malware surfaced in Ukraine in June after being spread by a malicious update to MeDoc, the country's most popular accounting software. Maersk picked up an infection that hooked into its global network and shut down the shipping company, forcing it to halt operations at 76 port terminals around the world.

Malware

- A generic term that covers various types of threats
- more and more complex attacks
- developed by criminal organizations, governments, companies…
- Effects
  - Blackmail
  - Spam relay
  - DDoS attacks
  - Data and identity theft, keylogger
  - Illegal file storage and sharing
  - Spread of malicious pieces of payload
  - Proxy malicious communications (to hide mothership)
  - Manipulation of polls, cracking of passwords…
- Data loss, system unavailability, financial loss…
- Recovery time!

Malware

- Malware propagates via vulnerability exploitation
- Vulnerability lifecycle (R. Anderson, Security Engineering, Wiley, 2008)

- Zero-day exploit
  - attack takes place as soon as vulnerability is discovered, before a patch is available
- Vulnerability (black) market (bug bounty)

Malware

- Trojan

Malware

- Trojan
  - program with undocumented features (often undesirable)
  - propagates via download, copy...
  - executed as a harmless program
  - mind your PATH variable
  - process inherits user’s privileges

Malware

- Virus
  - program that replicates by insert copies of itself into other programs
    - requires execution
  - 2 phases: insertion and execution
  - targets different components
    - boot sector, binary executable, macro
    - multipartite virus
  - can be resident or non-resident
  - encrypts and mutates to avoid detection
    - requires a decryption routine
  - often a multi-step propagation

Malware

- Worm
  - program that propagates across the network
  - Internet Worm, 1988
    - 3 propagation methods
      - rsh
      - finger buffer overflow
      - sendmail bug
    - once installed, attempts to break user passwords
  - Conficker (aka Downadup ou Kido)
    - Nov. 2008 → Mar. 2009: > 10M infected machines
    - uses known vulnerabilities (RPC) and unprotected shared disks
    - replicates, updates the registry
    - starts a local http server to allow copy distributions
    - creates a botnet
    - $250.000 bounty offered

Malware

- Spyware
  - 4 characteristics (Barwinski, 2006)
    - hidden
    - collects data (marketing, surveillance...)
    - sends collected data
    - resistant to removal attempts
  - Propagates through
    - infected downloads or toolbars
    - ActiveX controls
  - Effects
    - browser hijacking
    - advertisement
    - windows popping up all the time

Malware

- RootKit
  - programs and files hidden in the lower layers of the system (OS and below)
  - very resistant to detection
  - modifies the system in depth
  - various targets
    - hypervisor rootkit (blue pill, Rutkowska, 2006)
      - modifies the boot sequence to install an hypervisor first, that next runs the OS
    - kernel rootkit
      - most common
      - hidden as an OS component
    - library rootkit
      - hidden in a library (libc for instance)

Malware

- RootKit
  - detection methods
    - hash of critical files (keep hash safe)
    - start from external disk
    - monitor bahaviour and performance
    - monitor specific traces
    - ex: rkhunter
  - recovery?
    - try to cleanup: risky...is it really removed?
    - restore or re-install the system

Protecting against malware

- Mouse and cat game
- Prevention
  - select a secure OS and make it even more secure (OS hardening)
  - check software and data origin and integrity
  - filter email attachments, avoid active content,
  - block removable devices (USB, CD...)
  - apply proper system and process isolation
  - control access and privileges
  - train
- Detection
  - use a renowned anti-malware and keep it up-to-date
  - monitor system to detect unexpected changes in behavior
- Recovery
  - take regular backups
    - multiple versions
    - multiple copies/supports

OS Management

What protection features at OS level?

- authentication mechanisms
- access control to OS resources
- cryptographic primitives
- audit trail
- firewall
- IDS – Intrusion Detection System
- anti-malware
- vulnerability scanners

Resource management

- Isolation principle
  - easier resource management
  - easier resource protection
  - fine tune privileges
  - limit impact to one area
  - examples: Solaris Zones, chroot, sandboxing, cgroups, docker, virtualisation
  - use different environments for DEV, TST, QA, PROD
  - define migration procedures

Resource management

- User management
  - basis for proper access control and auditing
  - follow least privilege principle
    - account validity dates (not before, not after)
    - password rules
    - sandboxed environment
    - resource privileges
    - access restrictions (time, origin...)
  - avoid shared accounts
  - if needed, use privilege elevation or delegation (sudo, suid bit)

Resource management

- Resource management
  - control amount of resource used per user/process
  - CPU time, memory space, storage space, network bandwidth…
- Storage management
  - select appropriate filesystem
  - properly size volumes (volume full = full of troubles)
  - split data and system files (and logs…
  - control access
    - volume-level: ro, nosuid, noexec, quotas (hard/soft)
    - file/folder level: audit, ACL

Resource management

- Network security
  - local firewall
    - customize rules for IN/OUT traffic based on actual requirements
  - favor encrypted link ex: Secure Shell (ssh - http://www.openssh.org)

OS management

- Installation and configuration
  - initial setup
    - disk partitioning
    - network config
    - default passwords and policies
    - only install what is needed to minimize maintenance and risks
- Maintenance
  - patches and updates
  - major upgrades
- Documentation
  - preserve knowledge
  - easily rebuild in case of problem
  - operation and support documentation
- Training
- Contracts

Patches and updates

- Goal: patch vulnerabilities asap
- Apply necessary patches only
- Test patches before applying
- Define patching procedure
  - manual or automatic?
  - maintenance window?
- Major update
  - automatic or fresh install?

OS hardening

- limit the attack surface by limiting the amount of vulnerabilities
  - close all unnecessary doors
  - remove or deactivate unnecessary services and applications
  - configure firewall according to machine usage
  - install anti-malware
    - run a full system analysis
  - check the list of users
  - check file permissions (suid, root owned)
  - use encrypted channels for network services
  - synchronize the clock (Network Time Protocol – NTP)

Backup

Backup

- Define a backup plan
  - What?
    - various kinds of data (system, data, software, logs...)
  - How?
    - filesystem dump (snapshot), DB dump, file synchronization, dedicated tool
    - full or incremental?
  - When?
    - Frequency, retention period
    - limit impact on production
  - Specific infrastructure
    - storage, network
  - 3-2-1 rule
    - 3 total copies of your data, 2 of which are local but on different mediums, and at least 1 copy offsite.

Backup

- Backup security
  - protect access to media and devices
  - store securely
  - encrypt data
  - safe media disposal
  - LOCKSS – Lots Of Copies Keep Stuff Safe
  - validation
    - backup test
    - restore test

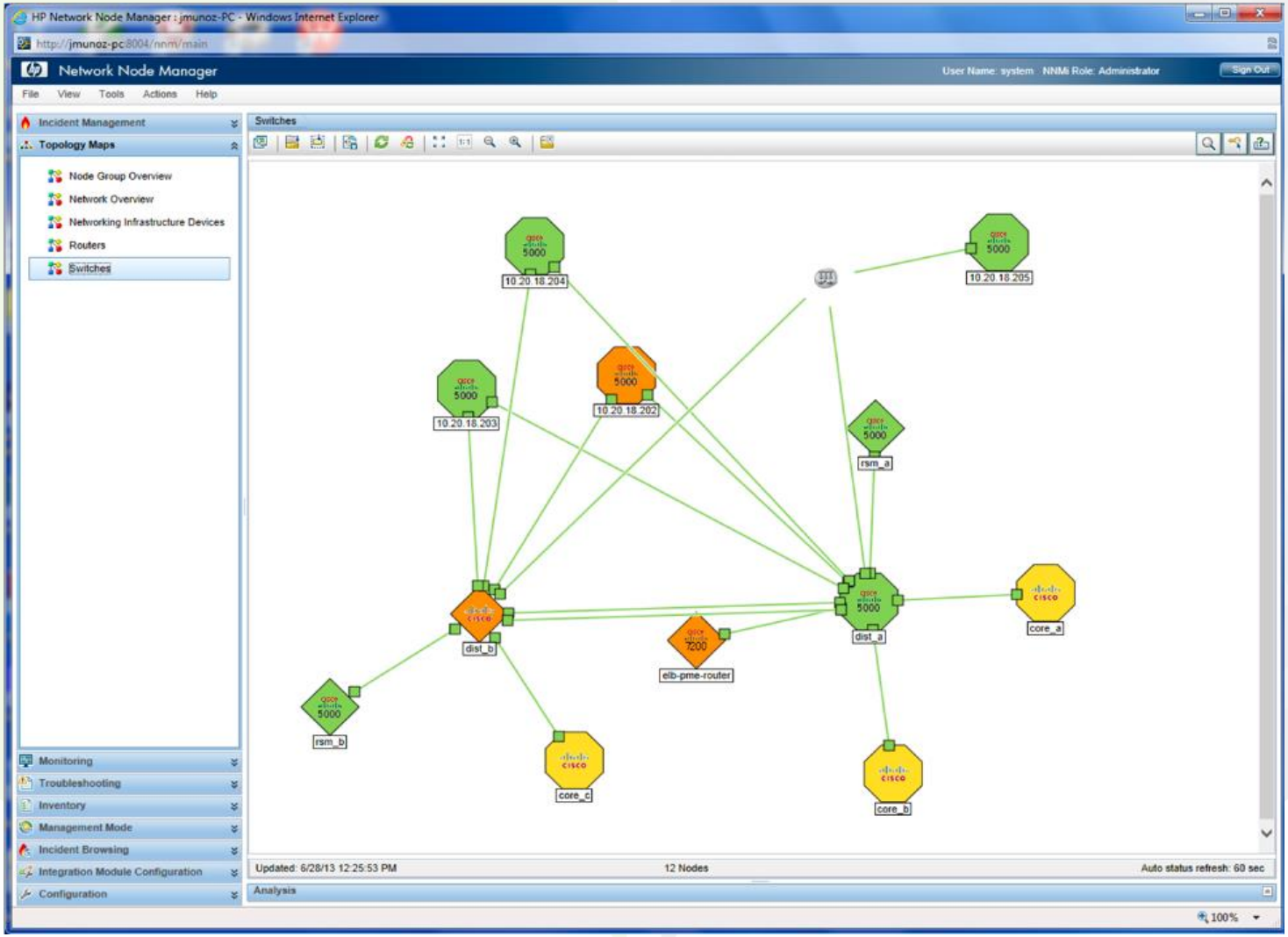
Monitoring and alerting

Monitoring and alerting

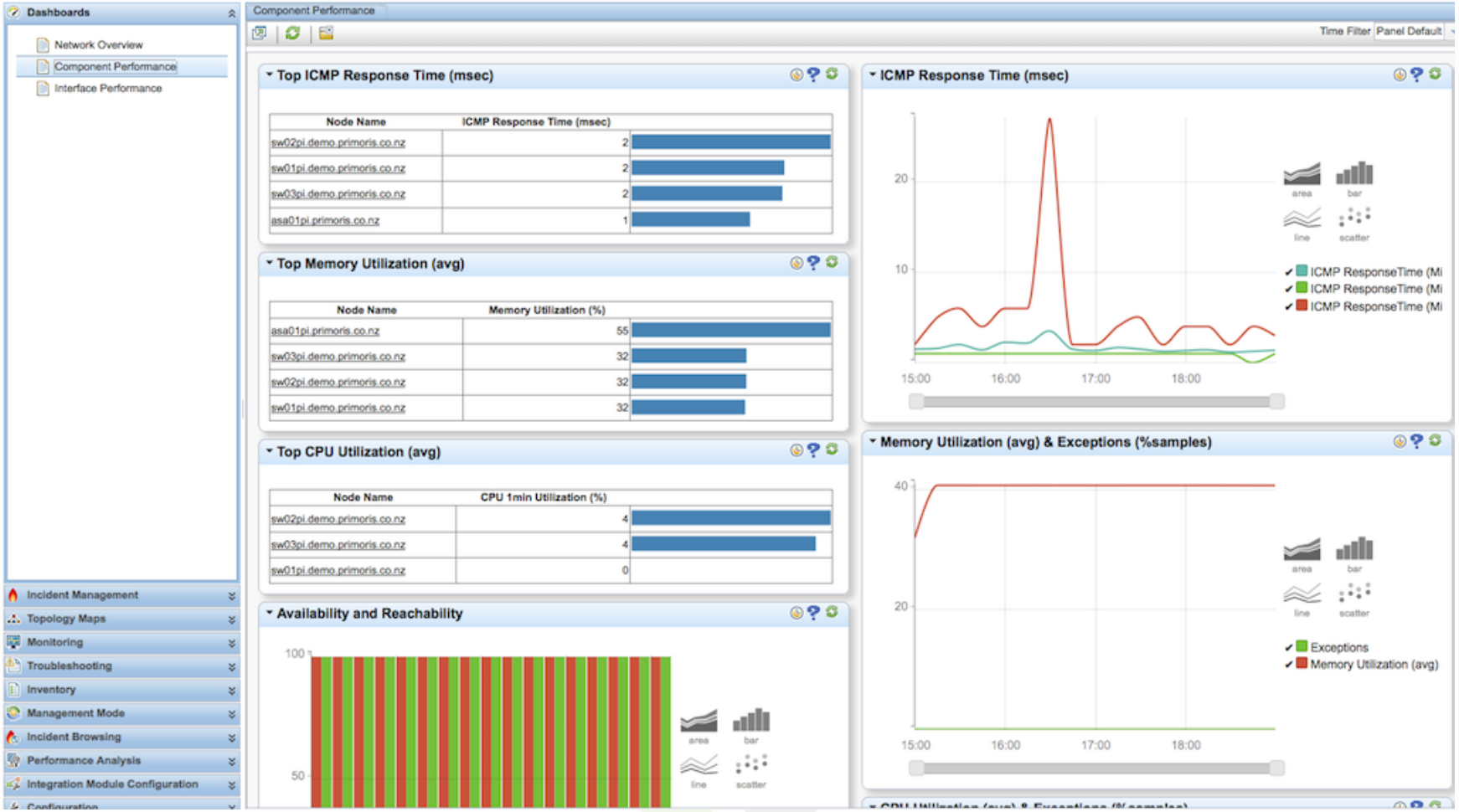
- Detection is critical, so system must be monitored
- In case of (possible) incident, response must be provided, which requires alerting, and available resources
- Monitor many different aspects
  - Availability, performance…
- Many tools exist
  - Local monitoring: system auditing
  - Network and configuration management
  - Opensource and commercial products
- Customization can be time consuming

Monitoring and alerting

Monitoring and alerting

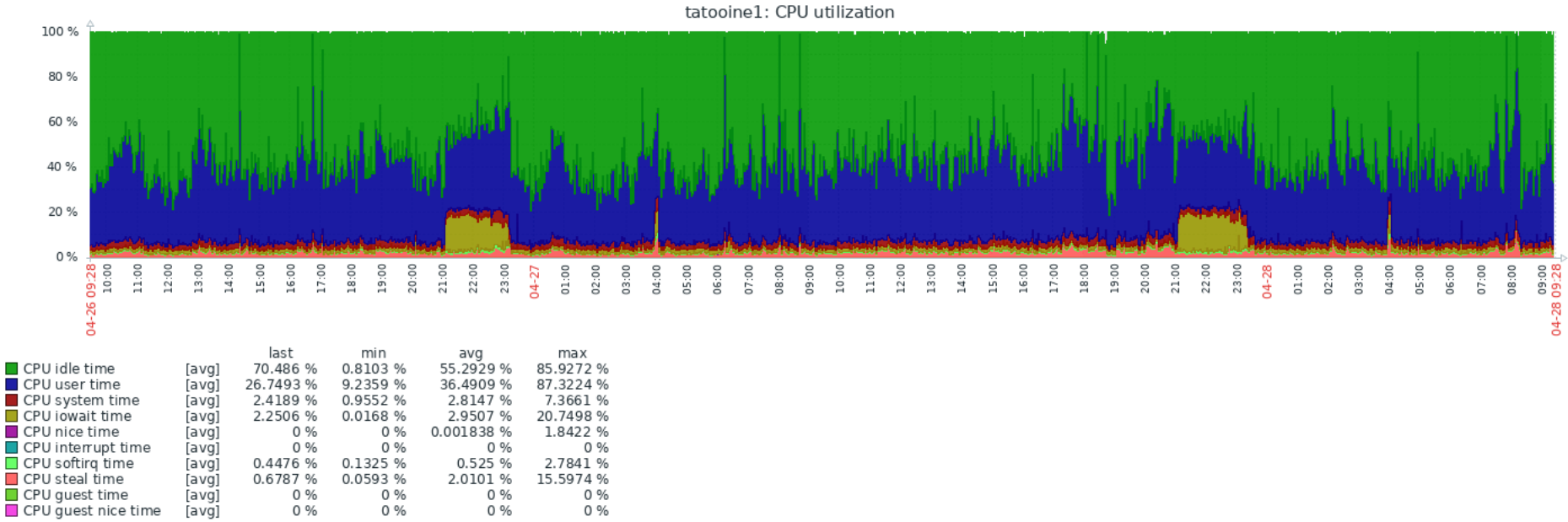


Monitoring and alerting



Monitoring and alerting

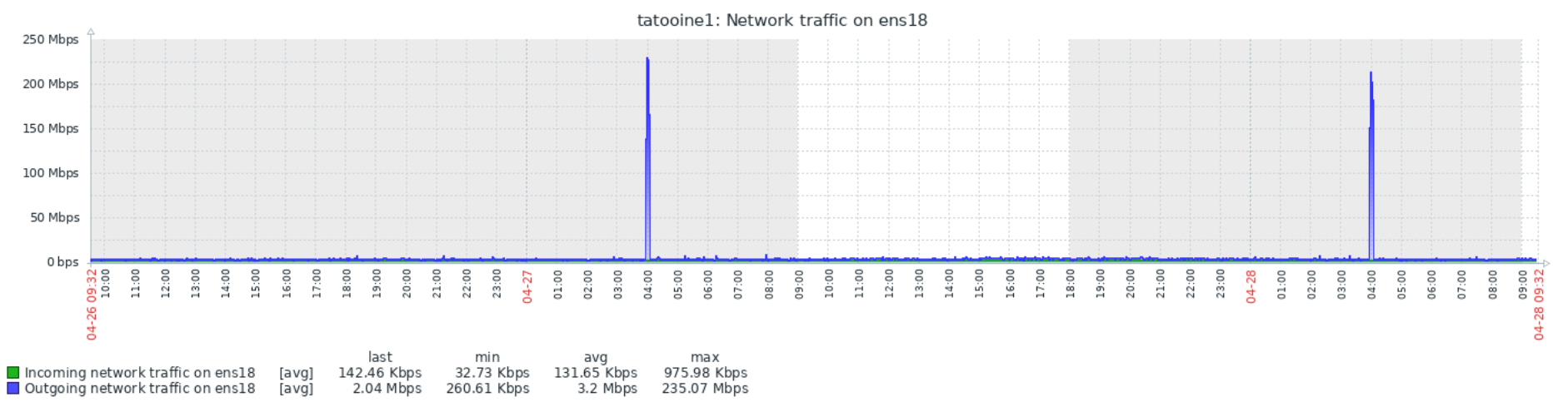
- How would you interpret this?



		last	min	avg	max
CPU idle time	[avg]	70.486 %	0.8103 %	55.2929 %	85.9272 %
CPU user time	[avg]	26.7493 %	9.2359 %	36.4909 %	87.3224 %
CPU system time	[avg]	2.4189 %	0.9552 %	2.8147 %	7.3661 %
CPU iowait time	[avg]	2.2506 %	0.0168 %	2.9507 %	20.7498 %
CPU nice time	[avg]	0 %	0 %	0.001838 %	1.8422 %
CPU interrupt time	[avg]	0 %	0 %	0 %	0 %
CPU softirq time	[avg]	0.4476 %	0.1325 %	0.525 %	2.7841 %
CPU steal time	[avg]	0.6787 %	0.0593 %	2.0101 %	15.5974 %
CPU guest time	[avg]	0 %	0 %	0 %	0 %
CPU guest nice time	[avg]	0 %	0 %	0 %	0 %

Monitoring and alerting

- How would you interpret this?



Security Information and Event Management – SIEM

- Goal: log management in the large
  - compliance reporting
  - log management
  - threat management
- collect, store, analyze, correlate all security-relevant events from all infrastructure components

Intrusion Detection System

- Goal: detect event that can reveal an intrusion
- Prevention is not enough; detection is also important
- 3 functions
  - gather and record data
    - distributed probes and sensors
    - carefully choose probes location
  - analyze data
    - normalize and process
  - detect intrusions
    - detect and alert

Intrusion Detection System

- Two approaches
  - Misuse detection
    - aka knowledge-based or signature detection
    - relies on a characterization of the attack (known bad)
    - system activity, network traffic, log messages...
    - signature database
    - most common approach
  - Anomaly detection
    - aka behavior-based detection
    - relies on a baseline
    - alert is generated in case measured behavior differs too much from baseline
    - able to detect new attacks
    - ‘new’ behavior can be considered as attacks
    - not easy to define the baseline

Intrusion Detection System

- Host based IDS
  - relies on local information
- Network based IDS
  - relies on network traffic analysis
- IDS reliability
  - false positive: costly and impacts trust in the system
  - false negative: security issue
  - requires fine tuning
- Limitations
  - often, attacks are crafted for specific system or software
  - signature database is constantly evolving
  - encrypted and/or fragmented traffic
  - substantial resources to analyze ⇒ risk of impact on infrastructure

SNORT (http://www.snort.org/)

- Signature-based multiplatform NIDS
  - Capture, decode, detect, output
- Rule
  - Action (Pass, alert, log), protocole, source, destination (IP et port)
  - Metadata (id, rev, msg, reference, type, priority)
  - Payload (content, nocase, uricontent, offset...)
  - Non-payload (low-level protocol flags: ttl, ipopts...)
  - Alert threshold

```
alert tcp $EXTERNAL_NET 1025: -> $HOME_NET any (msg:"MALWARE-CNC Vbs.Trojan.Agent inbound payload download"; flow:to_client,established; content:"s0|2D 7C 2D|"; fast_pattern:only; content:"Content-Length"; content:"s0|2D 7C 2D|"; within:200; metadata:impact_flag red, policy balanced-ips drop, policy max-detect-ips drop, policy security-ips drop, ruleset community; reference:url,blog.talosintelligence.com/2018/02/targeted-attacks-in-middle-east.html;
reference:url,virustotal.com/en/file/15f5aaa71bfa3d62fd558a3e88dd5ba26f7638bf2ac653b8d6b8d54dc7e5926b/analysis/; classtype:trojan-activity; sid:45643; rev:3;)
```

OSSEC (http://ossec.net/)

- Agent-based HIDS
- Multi-platform: Windows, most Unix flavors
- Features
  - rule-based engine
  - log file monitoring
  - file integrity check (content, owner, permission): hash-based, defined frequency
  - Windows registry check
  - rootkit detection (signature- and anomaly-based)
  - syslog support → incl. remote components (for instance Cisco routers, firewalls...)
  - correlation rules based on frequency and/or interval
- Alert
  - email, sms
  - stored in DB
- Active response
  - update firewall rules
  - disable user account
  - quarantine file
- Extensible and customizable
- Web Interface

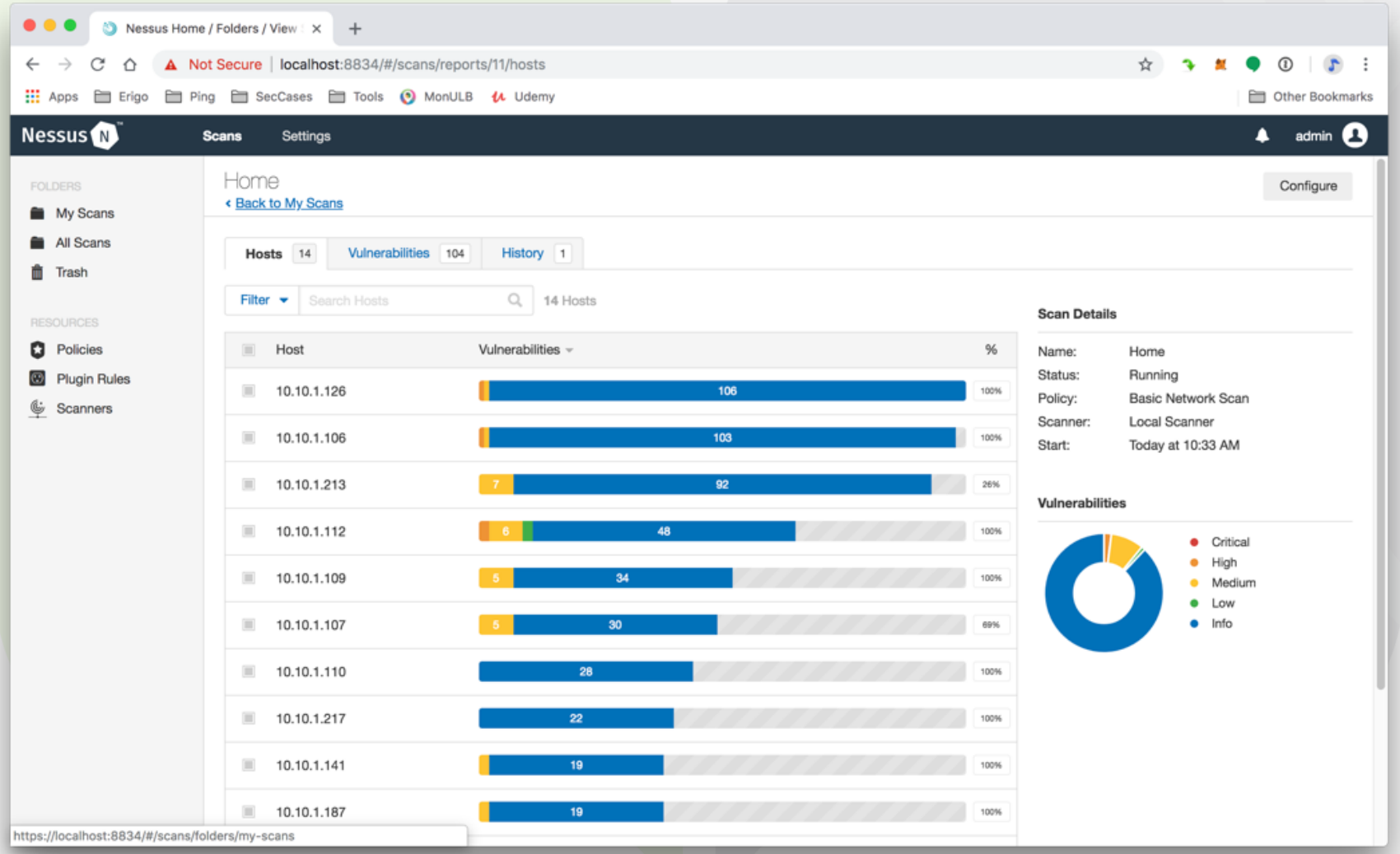
Vulnerability assessment

- Pro-active measure
- Automated tools
  - assess, patch, verify
  - detect and fingerprint system
  - detect and fingerprint applications and services
  - identify and report vulnerabilities
- Define aggressivity level: anticipate impact on the evaluation target
- Asset discovery, vulnerability scanning, misconfiguration, application flaws, malware, sensitive information identification...

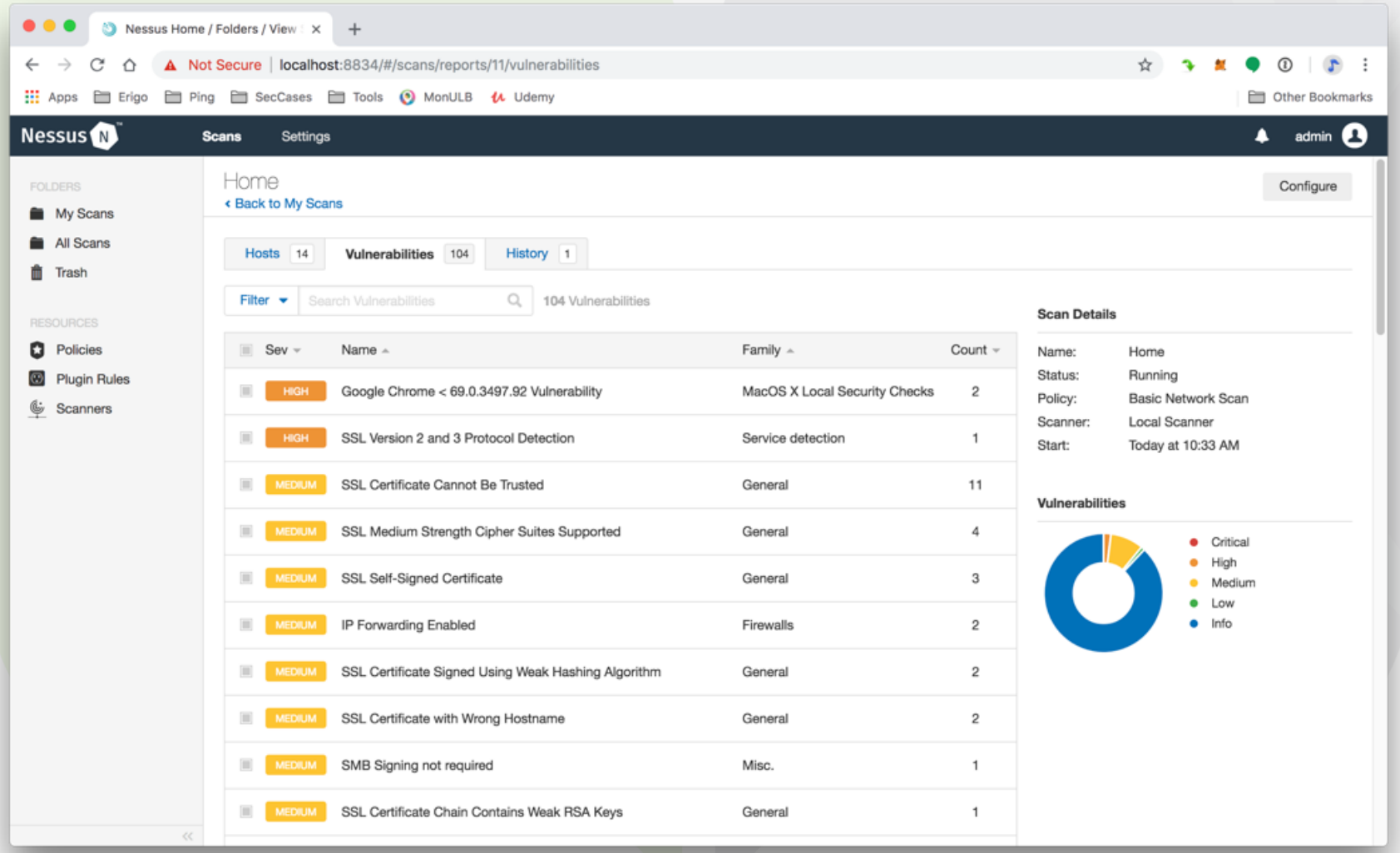
Vulnerability assessment

- Many tools available
  - Nmap (http://nmap.org/) network discovery, administration, and security auditing
  - Nessus (http: //fr.tenable.com/products/nessus-vulnerability-scanner) scans operating systems, network devices, next generation firewalls, hypervisors, databases, web servers and critical infrastructure for vulnerabilities, threats and compliance violations
  - OpenVAS (http://http://www.openvas.org/) open source vulnerability scanner
  - Metasploit (https://www.metasploit.org) complete pentest framework
  - LSAT – Linux Security Auditing Tool, Lynis, Tiger, BART – Basic Auditing and Reporting Tool modification to filesystem

Vulnerability assessment



Vulnerability assessment



Conclusion

Conclusion

- System software security requires action along the 3 lines
  - Prevention: isolation principle, OS hardening
  - Detection: monitoring, alerting, IDS/IPS
  - Recovery: backup
- Don’t forget support contracts

Some (hopefully) useful references

- ANSSI
  - Recommandation sur le nomadisme numérique (2018)
  - Recommandations relatives à l'administration sécurisée des SI (2018)
  - Recommandations pour choisir des pare-feux maîtrisés dans les zones exposées à Internet (2018)
  - Guide d'hygiène informatique (2017)
  - Définition d'une politique de pare-feu (2013)
  - Guide de définition d'une architecture de passerelle d'interconnexion sécurisée (2011)
- ENISA Threat Landscape Report https://etl.enisa.europa.eu
- ACM Queue: Security Collapse in the HTTPS market http://queue.acm.org/detail.cfm?id=2673311