

(Trusting ;-) Unauthenticated Protocols

An 2012 Overview of the Reality in Corporate
Environments



Motivation

What we see out there ...



EXERCISE
Feel the burn!!

- Almost all (corporate) environments we know ...
 - operate networks relying on unsecure/unauthenticated protocols.
 - have bad implementations, insecure configurations.
 - and don't really do much to somehow limit the possible impact.

What we have done ...

... and what we'll do.



- Look for protocols, that are...
 - Widely used in enterprises.
 - Important for security / compliance / proper operation.
 - Do not provide authentication or their authentication mechanisms are (mostly) unused.
 - If there is optional authentication, inspect some implementations and look if and how they use it.



First a little wrap-up from the past ...



TCP versus UDP communication

Regarding some attacks

Eavesdropping / Manipulation



- Attacks depending on a preceding eavesdropping or plain eavesdropping itself require man-in-the-middle position.
- TCP
 - Same network segment or MitM position required.
- UDP
 - Same network segment or MitM position required.

Source Spoofing



- TCP
 - Same network segment or MitM position required in order to spoof sender address and be able to receive responses.
- UDP
 - Anywhere in the network, as long as no spoofing protection or AC filtering is in place and answers are not required.

Injection (app level)



– TCP

- If IP source address is taken into account, same network segment or MitM position still required.

– UDP

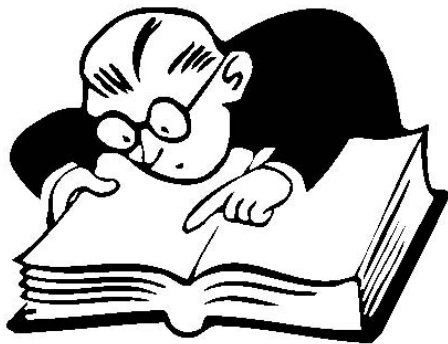
- Any place in the network, as long as no spoofing protection or ACL is in place.

Domain Name System

Of course ;-)

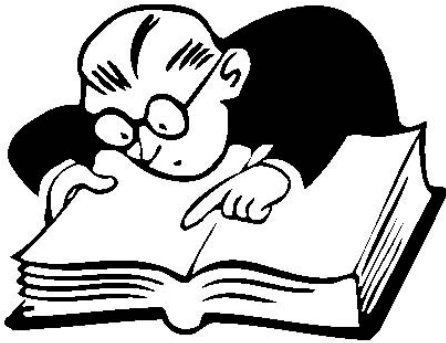


DNS Facts



- Sufficiently covered in the past! ;-)
 - So we won't do it again in detail.
- There are lots of tools available to mess with DNS.
- But what's the actual security impact in today's corporate environments?
- And how could the impact possibly be limited?

DNS Facts



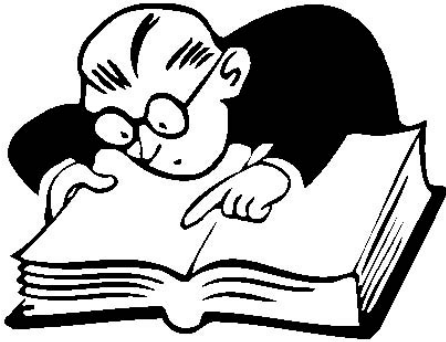
- Some facts:
 - Uses (mostly) UDP
 - (Simple) Manipulation of DNS requests requires MitM position (knowledge of request required)
 - Also cache poisoning (no MitM required) happened in the past.

DNS Facts



- What relies on DNS in typical corporate environments?
- It's not just client connections.
- Attackers can mess with basically every connection which isn't properly authenticated on application level or configured using IP addresses.

DNS Facts



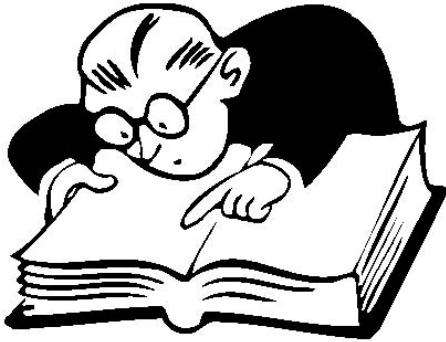
- Infrastructure critical resources are configured using DNS
 - Syslog Servers
 - NTP Servers
 - Network Management Stuff
 - (Active) Directory Servers

DNS Facts



- People/Systems making security critical decisions based on DNS resolution
 - Access Control Lists for e.g. databases
 - Download of configuration files
 - Download of network boot images
 - Access to certificate revocation lists

DNS Facts



- Untrustworthy DNS servers are used for resolution on ...
 - Whole networks
 - Or at least some systems

DNS Summary



- It is widely known that DNS is badly broken
 - Somehow all the administrators out there still seem to trust it.
 - In nearly every audit we find scary stuff on DNS.

I would like to see ...

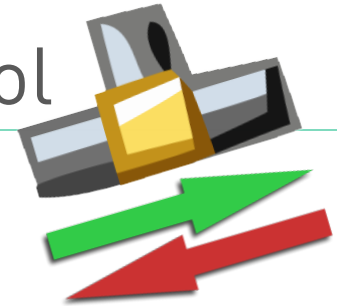
– That at least ...

- Trustworthy servers are used or better the resolution is handled by them selves.
- Critical resources with rarely changing addresses are configured by their IP address.
- Critical security decisions are not based on DNS resolution.



Simple Network Management Protocol

SNMP



SNMP Facts ...



- Also covered enough in the past, so no research required ! ;-)
- Also lots of tools available to mess with it (Also from ERNW)
- Still, the message seems not to have arrived at lots of IT stuff guys.

SNMP Facts ...

– SNMP is ...

- Used for monitoring / network management.
- READ, WRITE, TRAP operations
- Uses UDP (161/162)
- Authenticates using shared secrets



SNMP Facts ...

- SNMP is ...
 - V1 & V2 do not encrypt shared secret.
 - V3 rarely used / supported.
 - To get knowledge of shared secret, MitM position required (also for manipulation)
 - Often default values (public/private) used (See "Digging into SNMP in 2007")
 - => Also seen in every other pentest.



Back in 2007



Digging into SNMP in 2007 – An Exercise on Breaking Networks

**Enno Rey, erey@ernw.de
CISSP/ISSAP, CISA**

&

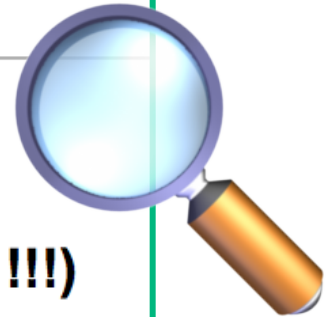
Daniel Mende, dmende@ernw.de



Back in 2007

Scanning the internet, some statistics

- **Of 240.000 *alive* addresses...**
- **~ 16.000 with SNMP “public”** (one out of 15 !!!)
- **~ 700 with SNMP “private”** (3 out of 1000)
- **=> in 350 million *alive* nodes approx 1.000.000 *privates***



SNMP Facts ...



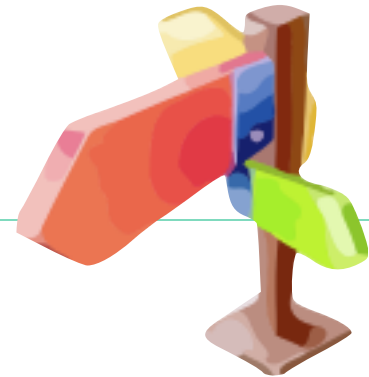
- The impact
 - Disclosure of critical information.
 - If write access available, configuration of network devices can be changed.
 - As SNMP Traps signal critical events, attackers can pretend critical states (hey, your primary uplink interface just went down)

I would like to see ...

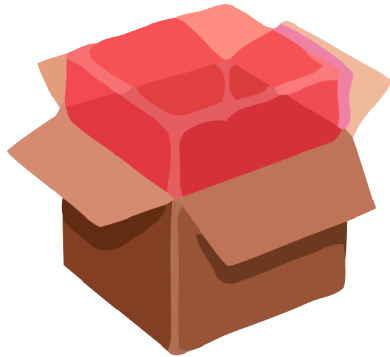
- That at least (even if v3 is not used)...
- Default community strings are changed to some \$COMPLEX_VALUE!
- \$MGMT_IF not reachable from \$CORP_NETWORK
- SNMP systems not configured using DNS, but IP ;-)



Routing protocols and Layer 2 stuff



Routing protocols / Low layer stuff



- There are lots of unauthenticated layer 2 protocols out there.
 - Multiple tools are available to mess with them (Loki, Yersinia, Ettercap, ...)
- Also most routing protocol setups do not use the offered authentication mechanism.
 - Or they have really simple values (like “cisco”)

See our 2010 BH talk on details

Loki supported protocols:

- ARP
- HSRP(v2)
- RIP
- OSPF
- EIGRP
- WLCCP
- VRRP(v3)
- BFD
- LDP
- MPLS,



The slide is a presentation poster for a talk at Black Hat USA 2010. The top banner is black with the ERNW logo (a green circle with a stylized 'E' and 'N' inside) and the text 'ERNW Living Security.' on the left. On the right of the banner is the 'black hat usa+2010' logo, which includes a stylized eye graphic and the text 'DATA. SELF. OFFENSE.' Below the banner, the slide is divided into two main sections. The left section features a large, fiery image of a cityscape at night, with the title 'Burning Asgard' in white text. The right section has a red background with the title 'Burning Asgard' in white, followed by a subtitle 'An Introduction to the Tool *Loki*' in white. Below the subtitle, the speakers' names 'Rene Graf, Daniel Mende, Enno Rey' are listed in white, followed by their email addresses '{rgraf, dmende, erey}@ernw.de' in white. At the bottom of the slide, there is a green bar with a small white logo consisting of a circle and a line.

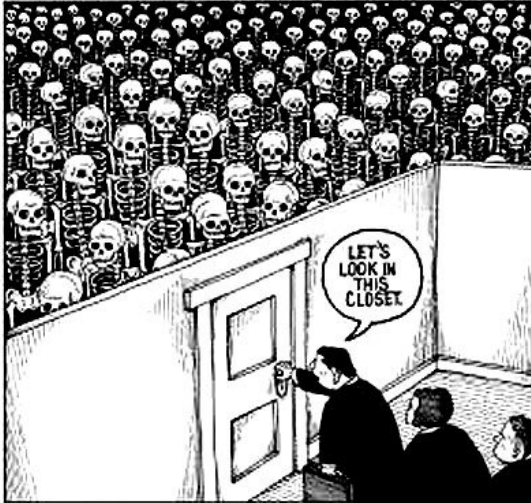
ERNW
Living Security.

black hat usa+2010
DATA. SELF. OFFENSE.

Burning Asgard
—
**An Introduction to
the Tool *Loki***

Rene Graf, Daniel Mende,
Enno Rey
{rgraf, dmende, erey}@ernw.de

Facts



- All corp environments “have their skeletons in the closet”.
- Try:
 - Connect to a network port
 - Fire up a sniffer
- → Seldom to don't see scary stuff.



Now some current research ...

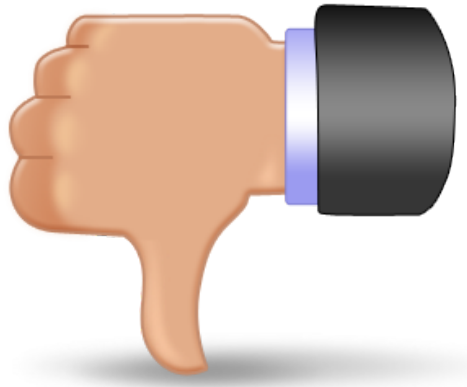
Stuff where no suitable tools to mess with it are available or which do not suite our needs

Trivial File Transfer Protocol

TFTP



Some facts ...



- VERY simple file transfer protocol
 - No authentication at all (except IP based ACL in some implementations)
- VERY old stuff
- But still found in nearly _every_ corp environment.

**Guess
when it first showed up ;-)**

Network Working Group
Request for Comments: 114
NIC: 5823

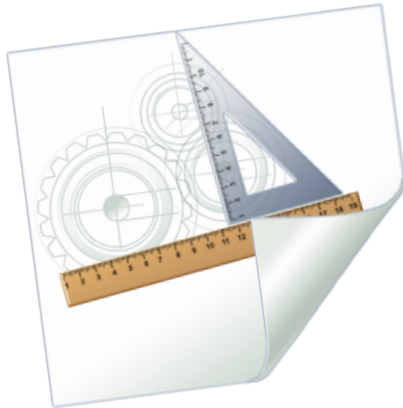
A. Bhushan
MIT Project MAC
16 April 1971

A FILE TRANSFER PROTOCOL

I. Introduction

Computer network usage may be divided into two broad categories -- direct and indirect. Direct usage implies that you, the network user, are "logged" into a remote host and use it as a local user. You interact with the remote system via a terminal (teletypewriter, graphics console) or a computer. Differences in terminal characteristics are handled by host system programs in accordance

The specs...



- First seen in RFC 114 (1971)
- Version 2 in 1981 (RFC 783)
- New revision in 1992 (RFC 1350)
- Since then: Multiple updates and add-ons.
- (RFCs: 1782, 1783, ..., 2349)

Usage



- Bootstrapping devices
 - First described in RFC 906 (1984)
(Bootstrap Loading using TFTP)
- Today still used for this purpose
 - Embedded stuff like VoIP phones
 - Firmware/configuration file download
- Configuration backup
 - E.g. Cisco: “write net”

Attack vectors



Target TFTP Server

- Download configuration files or other stored files (firmware images and such)
- → Knowledge of filename required.
- → Could be brute forced / guessed (often not that difficult)

Impact

- Information Disclosure
- Manipulation of files (=> broken integrity)

Attack vectors

- On the fly manipulation of TFTP transfers

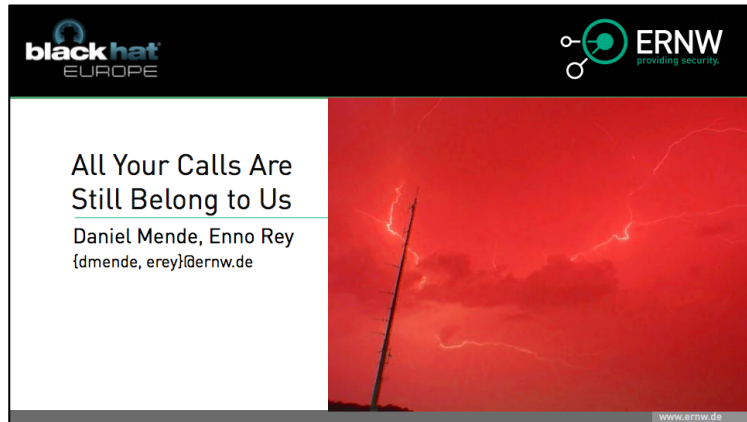
- Manipulate/Exchange/patch configuration file, firmware images or other downloaded files
 - Requires man-in-the-middle position
- Recently done in VoIP project.
 - Tool



New Tools

– TFTP Proxy

- Developed by Daniel during VoIP project to manipulate bootstrap process / break VoIP encryption of Cisco phones).
- To be released soon (Some adjustments needed to be usable in a universal manner)
- Watch our Blog (www.insinuator.net)



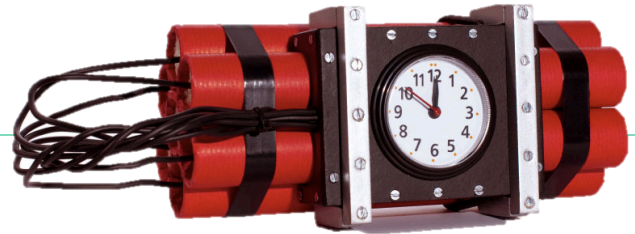
New Tools

– TFTP-BF

- Brute force available files on TFTP server
- To be released soon
- Watch our Blog (www.insinuator.net)



Network Time Protocol



Importance of correct system time



- A lot of critical functions rely on a correct system time
 - Certificate validation
 - Logging
 - Authentication mechanisms (E.g. Kerberos)
 - Cluster operation

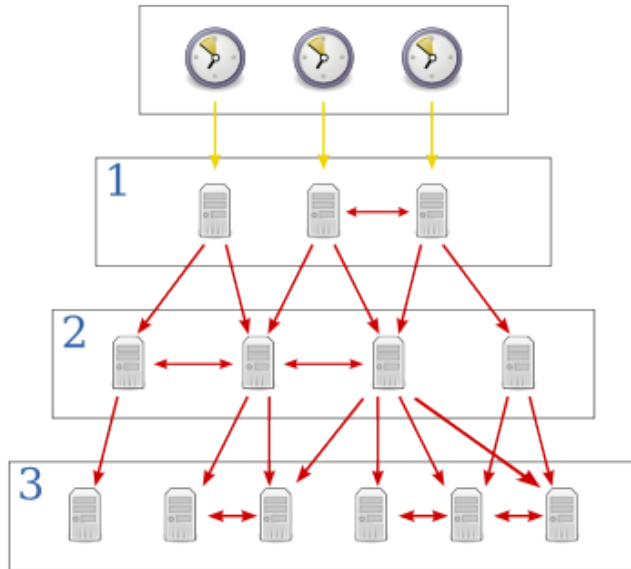
Some facts about NTP

An interesting one ...



- Used for system time synchronization by all platforms
- Uses UDP as transport
- Initially specified in 1985 (RFC 958)
 - No authentication mechanism available by then – Let's see if there are now.
- Multitude of newer specifications available now (security improved?)

Brief description

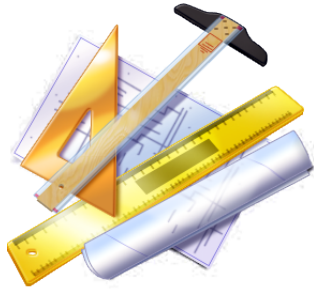


– Hierarchy

- Stratum 0 are reference devices like atomic clocks
- Stratum 1 systems are connected to Stratum 0 devices
- Stratum 2 systems sync with stratum 1 systems

– Peer synchronization performed for fault tolerance / error correction.

The specs



- RFC 958 in 1985 - NTP
 - “This RFC suggests a proposed protocol for the ARPA-Internet community, and requests discussion and suggestions for improvements.”
- RFC 1059 in 1988 – NTP v1
 - No authentication mechanism present
- RFC 1119 in 1989 – NTP v2
 - Authentication first suggested, but optional.

Authenticator (optional): When the NTP authentication mechanism is implemented, this contains the authenticator information defined in **Appendix C**.

Some more specs ...

- RFC 1305 in 1992 – NTP v3
 - Authentication still optional
 - Still widely used (E.g. Win7)

Authentication Variables

When the authentication mechanism suggested in Appendix C is used, the following state variables are defined in addition to the variables described previously. These variables are used only if the optional authentication mechanism described in Appendix C is implemented.

SNTP

- Simple NTP (SNTP)
 - For clients and “simple servers” (time not as exact as with full implementation, no state)
- RFC 1361 (1992), 1769 (1995), 2030 (v4, 1996), 4330 (v4, 2006)

Primary servers and clients complying with a subset of NTP, called the Simple Network Time Protocol (SNTPv4) [RFC4330], do not need to implement the mitigation algorithms described in Section 9 and following sections. SNTP is intended for primary servers equipped with a single reference clock, as well as for clients with a single upstream server and no dependent clients. The fully developed NTPv4 implementation is intended for secondary servers with multiple upstream servers and multiple downstream servers or clients. Other than these considerations, NTP and SNTP servers and clients are completely interoperable and can be intermixed in NTP subnets.

Latest specs ...



- RFC 9505 in 2010 – NTP v4
 - SNTP now included
 - Authentication still optional
 - Additional (new) authentication mechanism available as specified in additional RFC (we'll come back to that later)
 - Let's have a look at this (current) RFC

“15: Security Considerations”

- Oh, there is actually a security section. What does it say?

NTP security requirements are even more stringent than most other distributed services. First, the operation of the authentication mechanism and the time synchronization mechanism are inextricably intertwined. Reliable time synchronization requires cryptographic keys that are valid only over a designated time interval; but, time intervals can be enforced only when participating servers and clients

are reliably synchronized to UTC. In addition, the NTP subnet is hierarchical by nature, so time and trust flow from the primary servers at the root through secondary servers to the clients at the leaves.

**There is the first problem
with what you find in the real
world!**

Chain of trust



- What kind of NTP sources do you find in \$CORP_ENVIRONMENTS?
 - Some public servers, operators unknown.
 - Not that much environments have their own reference clock (Stratum 0)
 - Btw. Would this be more secure?



Chain of trust



- Also, the authenticity of the used NTP server must be verified.
- NTP has optional authentication since 1989 (NTPv2)

Ever seen NTP authentication in use?

“15: Security Considerations” - Continued



The NTP specification assumes that the goal of the intruder is to inject false time values, disrupt the protocol, or clog the network, servers, or clients with spurious packets that exhaust resources and deny service to legitimate applications. There are a number of defense mechanisms already built in the NTP architecture, protocol, and algorithms. The on-wire timestamp exchange scheme is inherently resistant to spoofing, packet-loss, and replay attacks. The

- So, there is something that should prevent injection of false timestamps without knowing the requests?
 - → We couldn't find this in the reference implementation.
 - → Actual OS implementations to be checked.

“15: Security Considerations” - Continued

scenarios. However, these mechanisms do not securely identify and authenticate servers to clients. Without specific further protection, an intruder can inject any or all of the following attacks:

- Nice, suggesting possible attacks; less work for us ;-)



“15: Security Considerations” - Continued

In a wiretap attack, the intruder can intercept, modify, and replay a packet. However, it cannot permanently prevent onward transmission of the original packet; that is, it cannot break the wire, only tell lies and congest it. Generally, the modified packet cannot arrive at the victim before the original packet, nor does it have the server private keys or identity parameters.

In a middleman or masquerade attack, the intruder is positioned between the server and client, so it can intercept, modify and replay a packet and prevent onward transmission of the original packet. However, the middleman does not have the server private keys.

- OK, relies on authentication. Again: Ever seen this?

Let's have a look at this authentication stuff ...

NTP Authentication

normative examples designed to illustrate the protocol's operation and are not a requirement for a conforming implementation. While the NTPv3 symmetric key authentication scheme described in this document has been carried over from NTPv3, the Autokey public key authentication scheme new to NTPv4 is described in [RFC5906].

- So, they copied the authentication mechanism from the 1992 version (NTPv3) - Which isn't necessarily bad.
 - And added something new (Autokey) – Sounds somewhat scary.

NTP Authentication

Key Identifier (keyid): 32-bit unsigned integer used by the client and server to designate a secret 128-bit MD5 key.

Message Digest (digest): 128-bit MD5 hash computed over the key followed by the NTP packet header and extensions fields (but not the Key Identifier or Message Digest fields).

- There's a key identifier which selects a locally stored key.
- And there's a message digest.
- Sounds not so bad (besides that MD5); What about this new Autokey stuff?

Latest specs ...



- NTPv4 Autokey: RFC 5906 (2010)
 - Specification to enable clients proof authenticity of NTP servers based on public key infrastructure.
 - → New thing, not supported by many public servers (ntp.org server list specifies this as entry property)
 - → Could solve problems, but I don't expect this to be widely used ever.
 - (remember, auth is available since 1989)

NTP Implementations



- ntp.org reference implementation
 - Mostly used in *NIX (ntpd, ntpdate, ...)
 - Supports NTPv4 including auth.
 - Autokey also supported.
- → FreeBSD, Archlinux, Ubuntu, OpenBSD

NTP Implementations



- OpenNTPD
 - Available in *NIX
 - Supports NTPv3 only.
 - No Authkey (NTPv4 Extension)

NTP Implementations



- Windows XP, Windows 7
 - Supports NTPv3 including auth.
 - Violates standard
 - No Authentication configuration.

```
Network Time Protocol (NTP Version 3, client)
  Flags: 0xdb
    11.. .... = Leap Indicator: unknown (clock unsynchronized) (3)
    ..01 1... = Version number: NTP Version 3 (3)
    .... .011 = Mode: client (3)
  Peer Clock Stratum: unspecified or invalid (0)
  Peer Polling Interval: 17 (131072 sec)
  Peer Clock Precision: 0.015625 sec
  Root Delay:      0.0000 sec
  Root Dispersion: 1.0156 sec
  Reference ID: NULL
```

NTP Attack Vectors



- Spoofing from MitM position
 - As suggested in RFC ;-)
 - Seems possible without problems (see packet sniffs)

NTP Request

```
▼ Network Time Protocol (NTP Version 4, client)
  ▼ Flags: 0xe3
    11.. .... = Leap Indicator: unknown (clock unsynchronized) (3)
    ..10 0... = Version number: NTP Version 4 (4)
    .... .011 = Mode: client (3)
  Peer Clock Stratum: unspecified or invalid (0)
  Peer Polling Interval: 6 (64 sec)
  Peer Clock Precision: 0.000000 sec
  Root Delay:      0.0000 sec
  Root Dispersion: 0.0000 sec
  Reference ID: (Initialization)
  Reference Timestamp: Jan  1, 1970 00:00:00.000000000 UTC
  Origin Timestamp: Jan  1, 1970 00:00:00.000000000 UTC
  Receive Timestamp: Jan  1, 1970 00:00:00.000000000 UTC
  Transmit Timestamp: Oct  4, 2012 20:13:57.749536000 UTC
```

NTP Response

```
▼ Network Time Protocol (NTP Version 4, server)
  ▼ Flags: 0x24
    00.. .... = Leap Indicator: no warning (0)
    ..10 0... = Version number: NTP Version 4 (4)
    .... .100 = Mode: server (4)
    Peer Clock Stratum: secondary reference (2)
    Peer Polling Interval: 6 (64 sec)
    Peer Clock Precision: 0.000002 sec
    Root Delay:      0.0237 sec
    Root Dispersion: 0.0143 sec
    Reference ID: 192.53.103.103
    Reference Timestamp: Oct  4, 2012 20:05:47.156642000 UTC
    Origin Timestamp: Oct  4, 2012 20:13:57.749536000 UTC
    Receive Timestamp: Oct  4, 2012 20:10:57.516070000 UTC
    Transmit Timestamp: Oct  4, 2012 20:10:57.516121000 UTC
```

NTP Attack Vectors

- Spoofing from wiretap position
 - As suggested in RFC ;-)
 - Seems possible
 - Probably DoS of NTP Server required or by flooding with huge amount of packets.



NTP Attack Vectors



- Blind spoofing without knowledge of requests.
 - Knowledge of request timestamp possibly required.
 - → Implementation testing still work in progress. We have Dizzy script ready (ERNW Fuzzing Tool)

NTP Attack Vectors



- An Auto discovery mechanism
→ To be researched
- Broadcast / Multicast Time Sync
→ To be researched

SYSLOG



SYSLOG



- Also an interesting one ...
- Used for local and central logging on *NIX systems, network devices
- Agents for Win available
- Mostly used over UDP, TCP also available in some implementations.

SYSLOG



- What we see often ...
- Logging and traceability of events very important because of compliance reasons.
- But implemented using syslog!

The Specs



- First RFC in 2001 (RFC 3164)
 - Before that only de facto standard
- Current Spec is 5424 (2009)
- Lot's of incompatible formats out there. → Parsing sucks.

The Specs



- New additions
 - RFC 5425 – TLS Support (2009)
 - RFC 5426 – Syslog over UDP (2009)
 - ...
 - RFC 5848 – Signed Messages (2010)
- But again: Never seen in the wild!

Impact

- Fake syslog messages can have serious consequences ;-)

```
Oct 12 11:31:29 mx1 postfix/smtpd[21226]: 59A2215EC2E: client=unknown[172.31.1.10]  
Oct 12 11:31:29 mx1 postfix/cleanup[21277]: 59A2215EC2E: message-id=<964E7044-B50E-492D-9D72-82569F08  
Oct 12 11:31:29 mx1 postfix/smtpd[21226]: disconnect from unknown[172.31.1.10]  
Oct 12 11:31:29 mx1 postfix/qmgr[81566]: 59A2215EC2E: from=<vip@someagency.gov>, size=8587, nrcpt=1 (C  
Oct 12 11:31:31 mx1 postfix/smtp[21278]: 59A2215EC2E: to=<yoursecretsplease@wikileaks.com>, relay=mai  
Oct 12 11:31:31 mx1 postfix/qmgr[81566]: 59A2215EC2E: removed
```

- Think anybody will question this?

Impact



- What about injecting some proxy logs showing your unloved coworker browsing porn all the time?
 - And then filing a complain.
- Think anybody will ask if the logs are authentic?

Impact



- Or just simple things like letting a whole network appear completely broken – although there isn't any problem.
- Have your co-admin come in every night at 3am ;-)
- And of course: Completely broken compliance requirements.

Attack vectors



- Syslog filtering from MitM position
- Log injection from any position in \$CORP_ENVIRONMENT
 - As long as no spoofing protection is in place (e.g. on firewall device). But: Who does filtering in the internal network?

Next steps

- Finish development of SYSLOG-Proxy.
- Finish development of Syslog injection tool.
 - Scapy can do this.
 - We'll provide a tool which has profiles for different fake events.
- Both are still work in progress.



How to fix it



- Use Syslog over TCP / SSL
- When coming available, use new extensions (TLS, Signing, ...)
- Filter Traffic
- Isolate log systems from client networks
- BTW: Don't use DNS for server config ;-)

Conclusion



To sum it up ...



- There are still too many bad protocols out there.
- They expose serious threats to your environments.

What to do ... To defend



- Be aware of the risks associated with those protocols.
- Do not try to fulfill compliance requirements with plain syslog.
- Replace them where possible.
- Do use trusted / verified sources (DNS, NTP)
- Configure them properly where possible.
- Isolate them from potential attackers.

What to do ... To attack

- Watch ... for tool releases and updates:
 - ERNW Website: <http://www.ernw.de>
 - ERNW Newsletter
 - ERNW Blog: <http://www.insinuator.net>
- Visit Troopers13 in Heidelberg / Germany from 11th to 15th March 2013.





Questions & Answers
