

Automated Signature Generation: Overview and the NoAH Approach

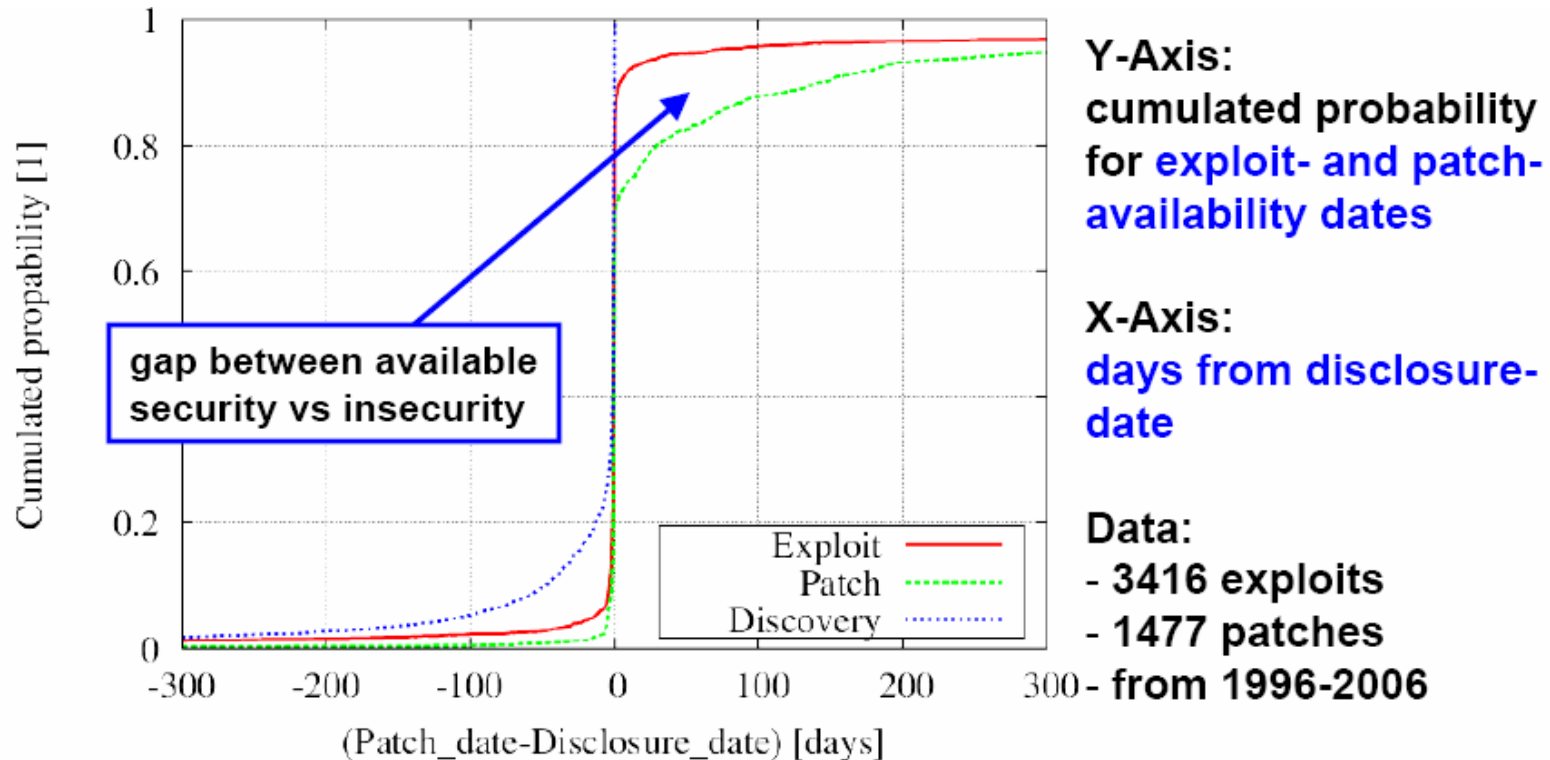
Bernhard Tellenbach



Structure

- Motivation: The speed of insecurity
- Overview
 - Building Blocks and Techniques
- The NoAH approach

The speed of insecurity



Source: "The Dynamics of (In)Security", Stefan Frei, ETH Zurich, BlackHat 2006

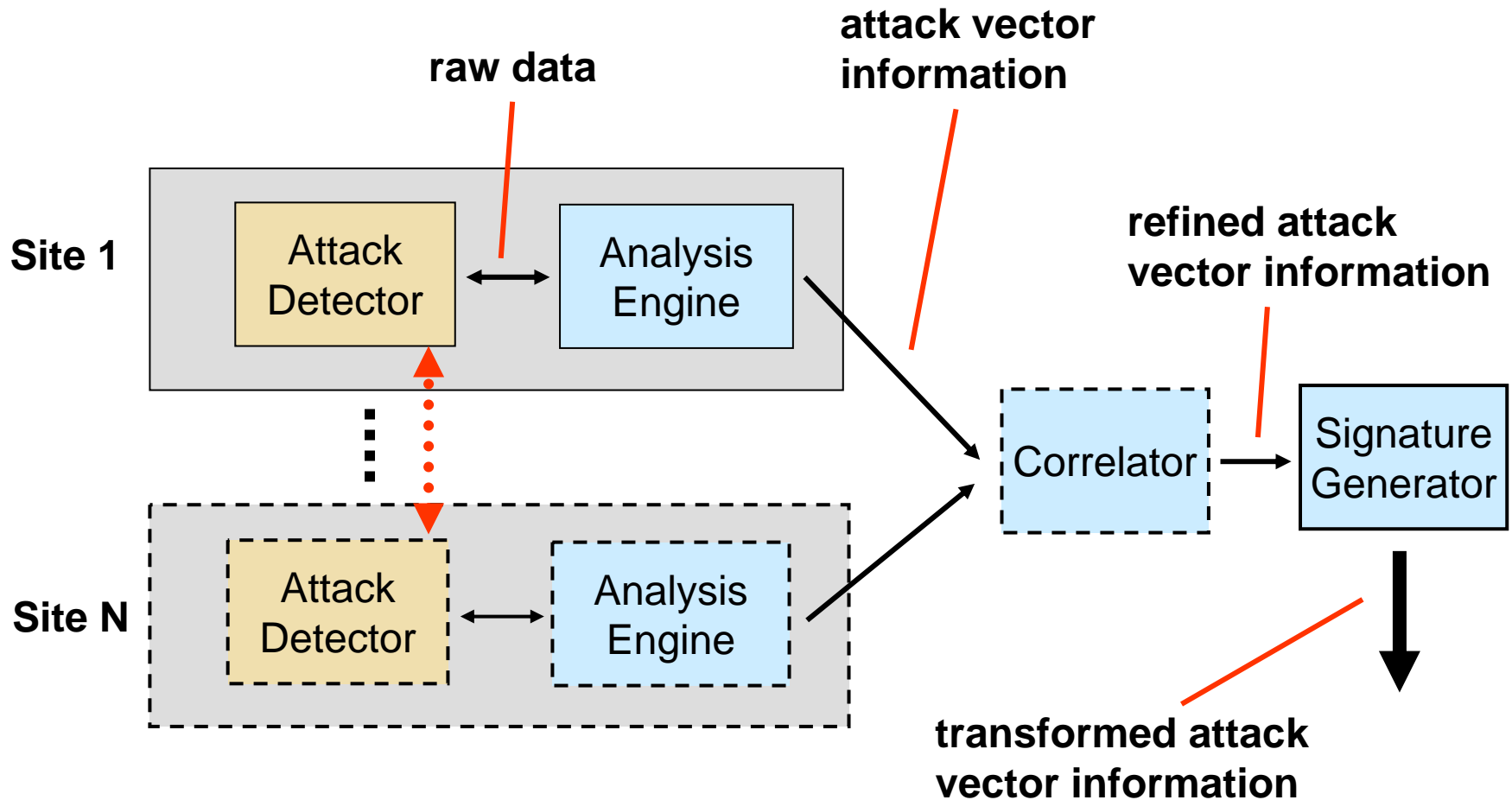
Motivation

- Disclosure and Exploit are (sometimes) too close
 - > Zero-Day exploits are a fact
 - > Manual SG is too slow



- Automated Signature Generation (ASG) could...
 - ...make security products cheaper
 - ...render remote updates of signature databases obsolete

Overview: Building Blocks of an ASG System

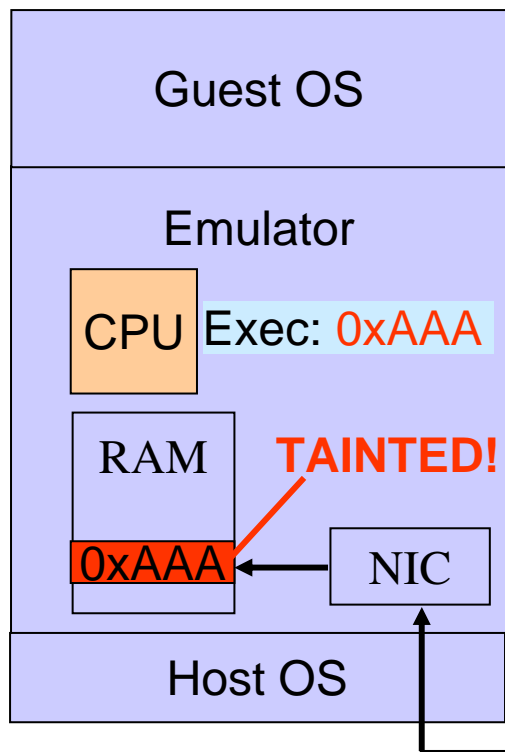


Techniques for Detecting Unknown Attacks

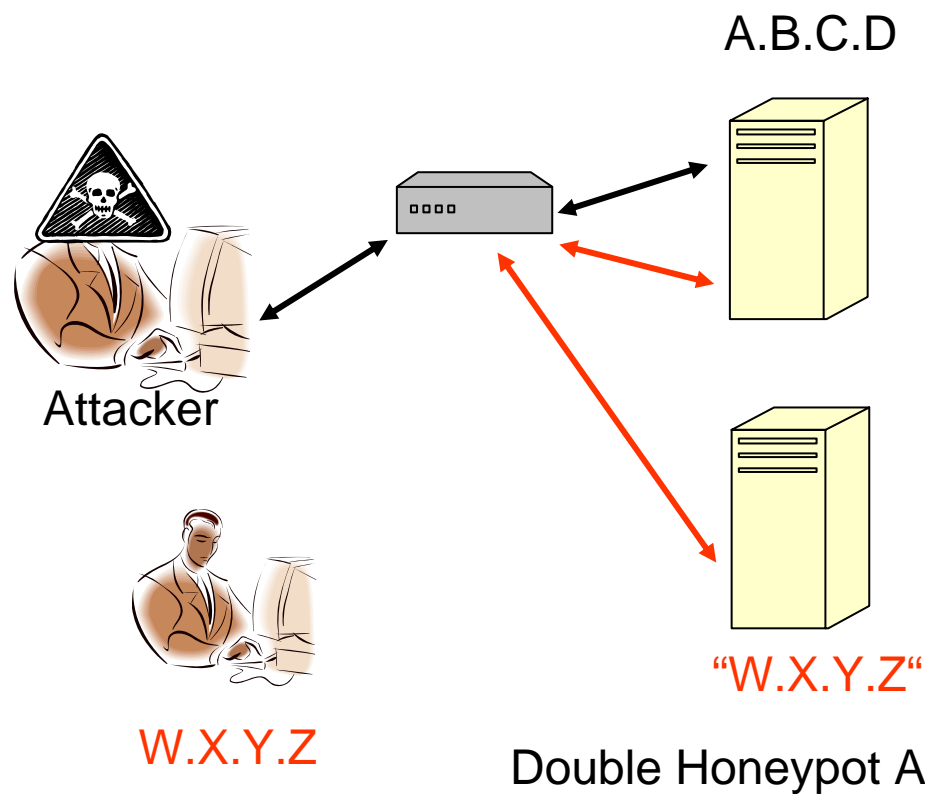
	Anomaly based	Compromise based
Standalone	Host-Based <ul style="list-style-type: none"> • Execution path profiling [1] • Traffic to honeypot [2] Network-Based <ul style="list-style-type: none"> • Profiling of benign traffic [3] 	Host-Based <ul style="list-style-type: none"> • Address Space Randomization [6] • Memory tainting [7] Network-Based <ul style="list-style-type: none"> • Honeypot [4] • (Protocol violations)
	Distributed	Network-Based <ul style="list-style-type: none"> • Scanning/Spreading [5]

Compromise Based Detection Techniques

Memory Tainting:



Honeypot:



Techniques for Detecting Unknown Attacks

■ Limitations:

- **Anomaly based techniques:** Are prone to false positives and false negatives
- **Compromise based techniques:** Mainly for buffer overflow and/or code injection attacks

■ Ready for use in production infrastructures?

- All techniques can be found “in the wild” (in software and/or hardware)

Techniques for Network Traffic based Analysis Engines

- **Group traffic** (e.g. according to service) and:
 - [3] search for longest common substring (LCS)
(> only for non-polymorphic attack payload)
 - [8] search for characteristic n-grams
(> can handle attack payload with block reordering)
 - [4] extract significant regions using position aware byte frequency analysis
(> can handle attack payload with limited polymorphism)



Output: Byte sequence(s) or byte frequency distribution

Techniques for Network Traffic based Analysis Engines

■ **Limitations:**

- Requires MULTIPLE SAMPLES of the “same“ attack
- Are prone to false positives and evasion:
 - Systematic manipulation of byte distributions
 - Injection of fake characteristic byte sequences
- Only for unencrypted traffic

■ **Ready for use in production infrastructures?**

- No. At least not for use in IPS
- Could be used as add-on to manual analysis

Techniques for Host Information Based Analysis Engines

- Memory analysis: Identify relevant content
 - Log memory content injected by the attacker
 - Log its use (data, code, jump address,...)
- How?
 - Instrumentation of applications
 - [9] E.g. source code or binary rewriting
 - Application independent
 - [10] OS independent (e.g. x86 emulator based)
 - [11] OS dependent (e.g. using memory debuggers like Valgrind)



Output: Memory addresses and content relevant for the attack

Techniques for Host Information Based Analysis Engines

- Execution path analysis:
 - Identify path characteristics toward the vulnerability
 - Identify modifications required to exploit vulnerable code
- How?
 - Instrumentation of applications
 - [1] E.g. source code or binary rewriting
 - Use tools based on debugging techniques



Output: Charact. exec. flow and data required to control it

Techniques for Analysis Engines

■ Limitations:

- Performance: 2 to 100 times slower
- Online analysis: Fingerprinting of sensors is “easy”
 - Mitigation: Offline analysis > Requires reliable REPLAY techniques!
- Limited to buffer-overflow/code-injection attacks

■ Ready for use in production infrastructures?

- Yes, especially as add-on to conventional IDS
- Use honeypot based systems with almost no maintenance overhead (e.g. [4])

Techniques for Analysis Engines

- Hybrid analysis engine:
 - Host information and network traffic based
> **The NoAH approach**

Techniques for other components:

- (Correlation)
- Signature Generator:
 - Translate information from analysis engine into a signature

The NoAH Approach

EU project NoAH (Network of Affined Honeypots)

Goals

- NoAH aims at automated
 - detection of unknown attacks
 - Generation of signatures to counter 0-day attacks
- **Generate signatures for common IDS**
- Install full-scale infrastructure across Europe
- Target audience: ISP's, NREN's, researchers

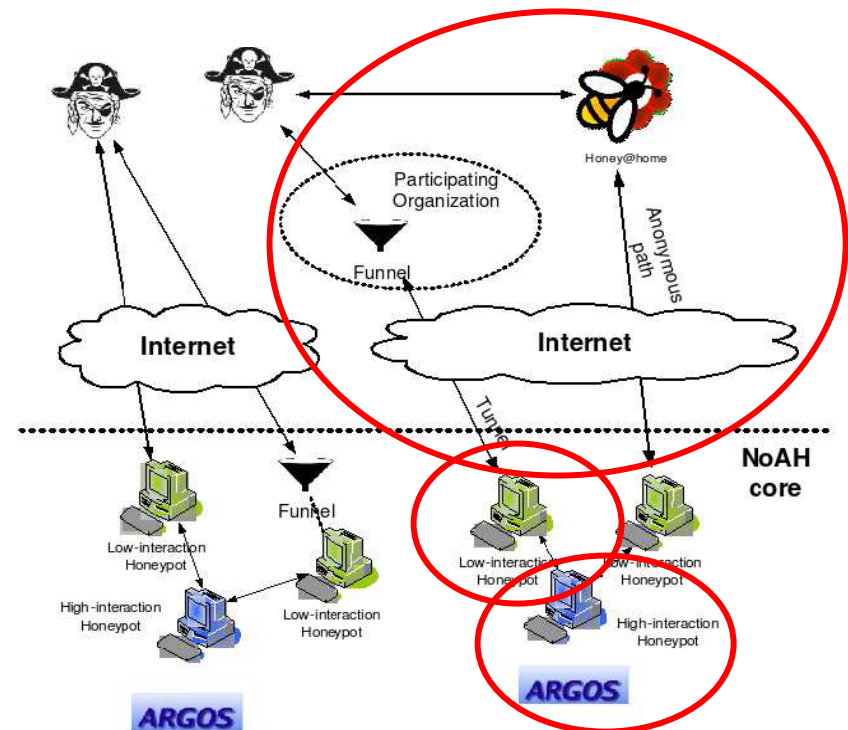
Goal: Generate signatures for common IDS

- Most IDS use network traffic based signatures
- We chose SNORT signatures because...
 - SNORT is Open source and well-known
- Implications:
 - **False positives are likely to be a problem**
 - **No useful signatures for encrypted traffic**

Attack Detection

NoAH Architecture: Attack Detection Overview

- Traffic to unused Internet addresses is redirected
- Low-interaction honeypots filter unwanted traffic
- High-interaction honeypots form the attack detectors (Argos)
- ASG is (partially) integrated with Argos



NoAH Architecture: Attack Detector Argos

- Detection technique (Argos):
 - OS independent memory tainting (x86 emulator)
 - Mark all data from the network interface as tainted
 - Attack is reported if:
 - tainted memory regions are used as jump address
 - virtual CPU executes code from memory location that is tainted
 - Detects code injection attacks with almost 100% accuracy by design

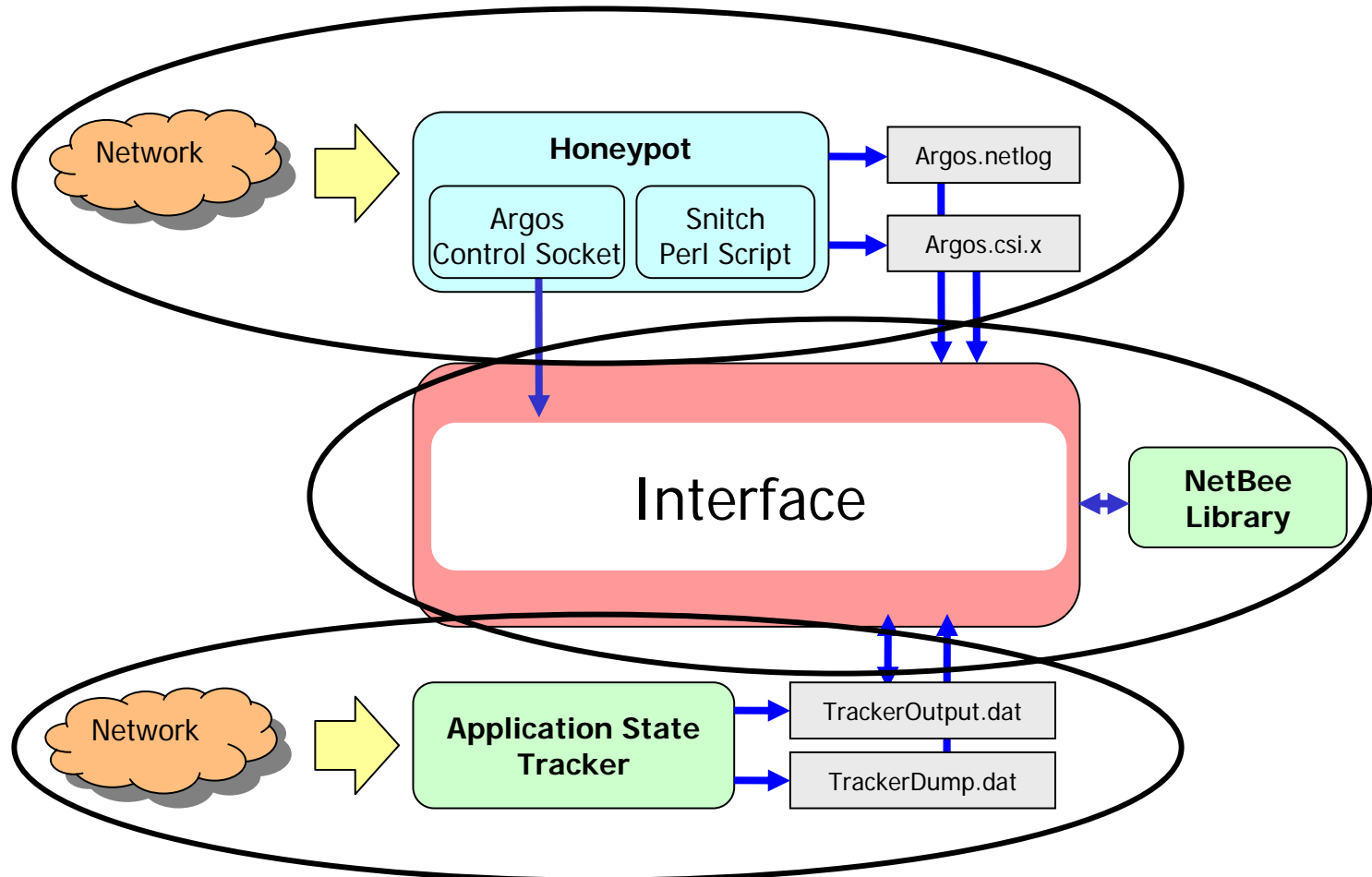
> Scope of NoAH: Remote attacks that do not require a human in the loop

Attack Analysis

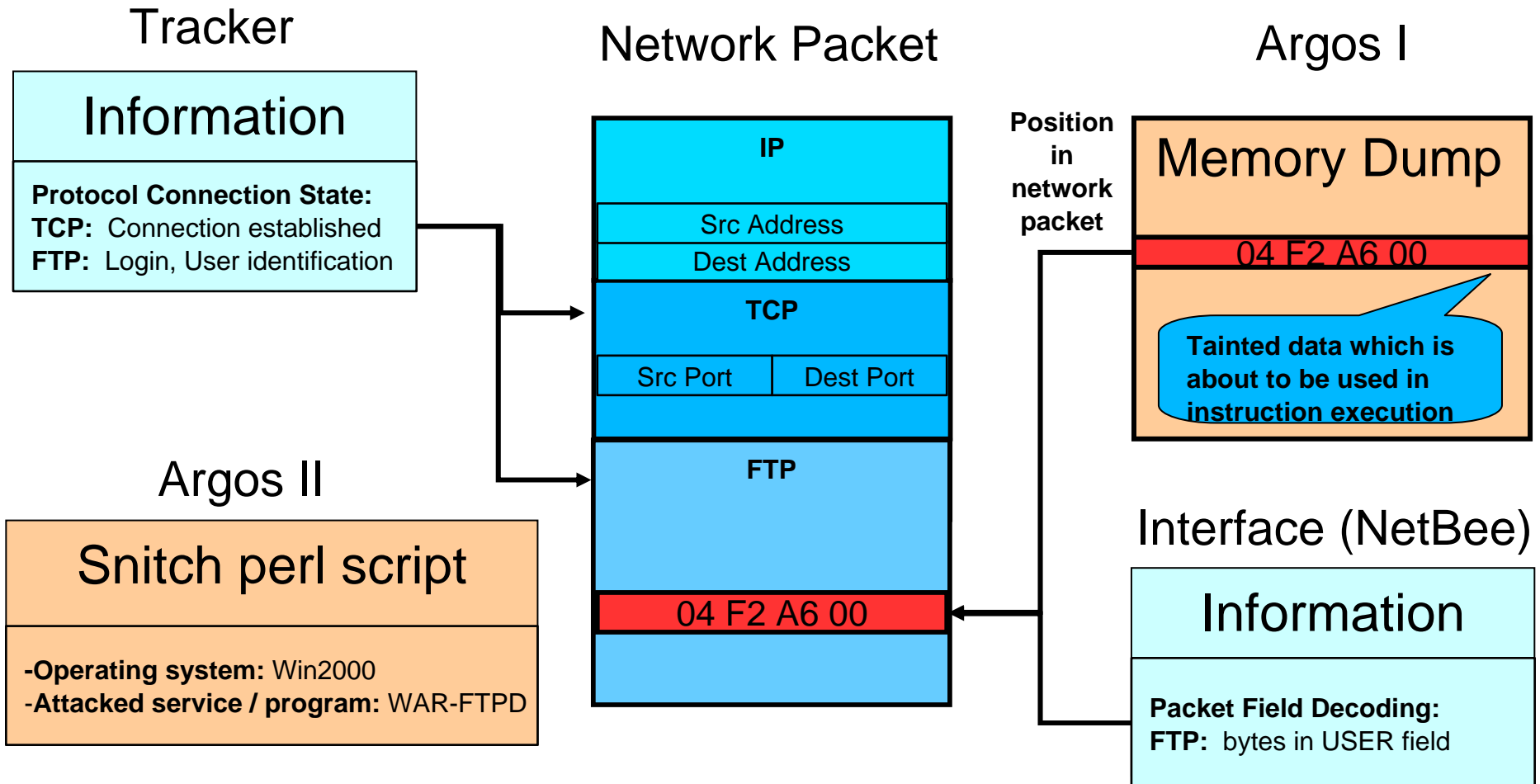
Hybrid Analysis Engine: Idea

- Identify memory content relevant for the attack
- Identify network traffic bytes involved
 - Log all memory operations to keep track of the origin
 - Only bytes relevant for the attack are used for SG
- Extract meta information:
 - Protocol field(s) containing the relevant bytes
 - Make use of the NetBee Library (support for 64 protocols) [12]
 - Communication/Protocol state history
 - Extensible tracker framework
 - Proof-of-concept implementation to track (IP,TCP/UDP,FTP)

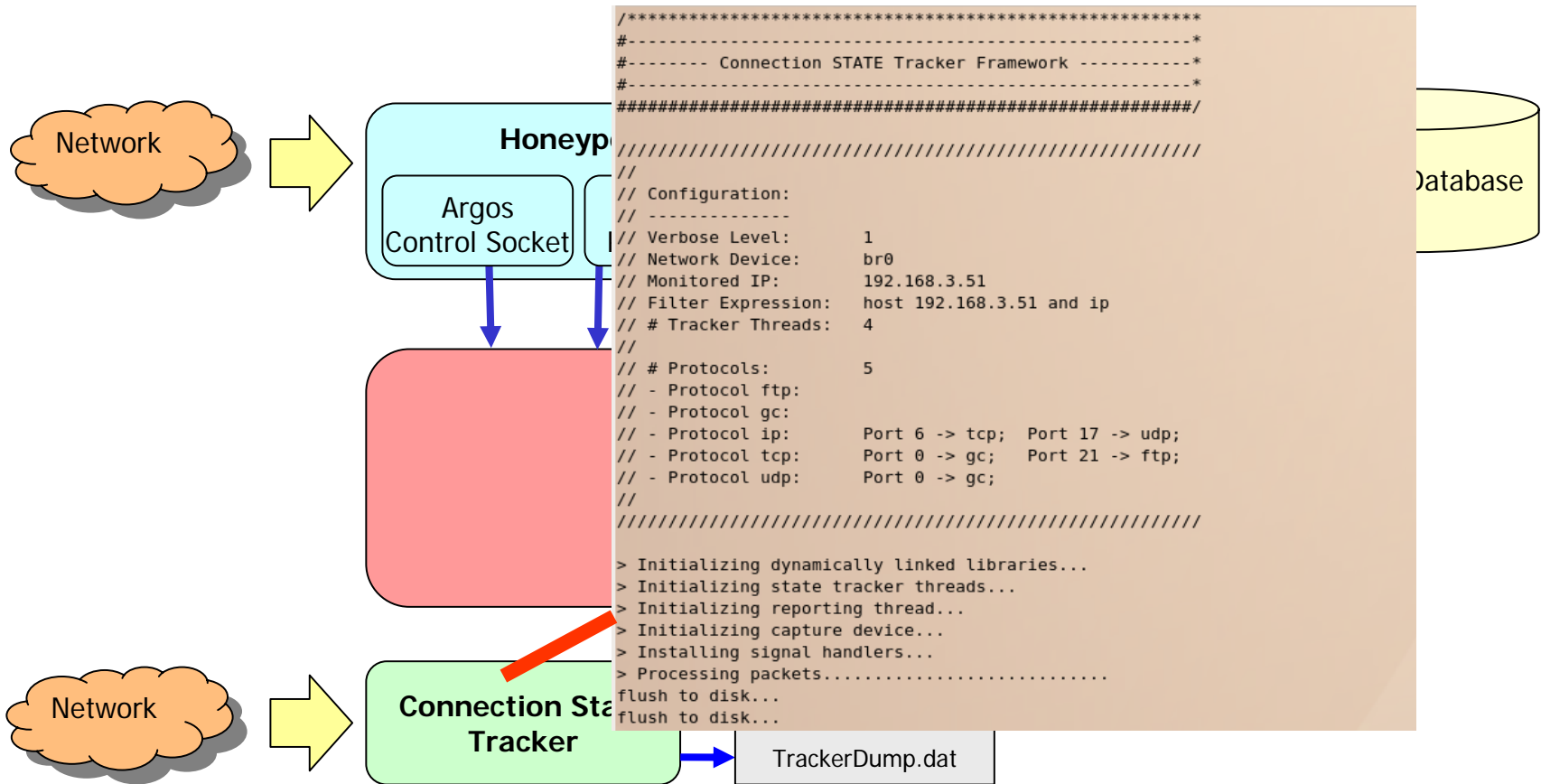
Basic Architecture of the entire ASG System



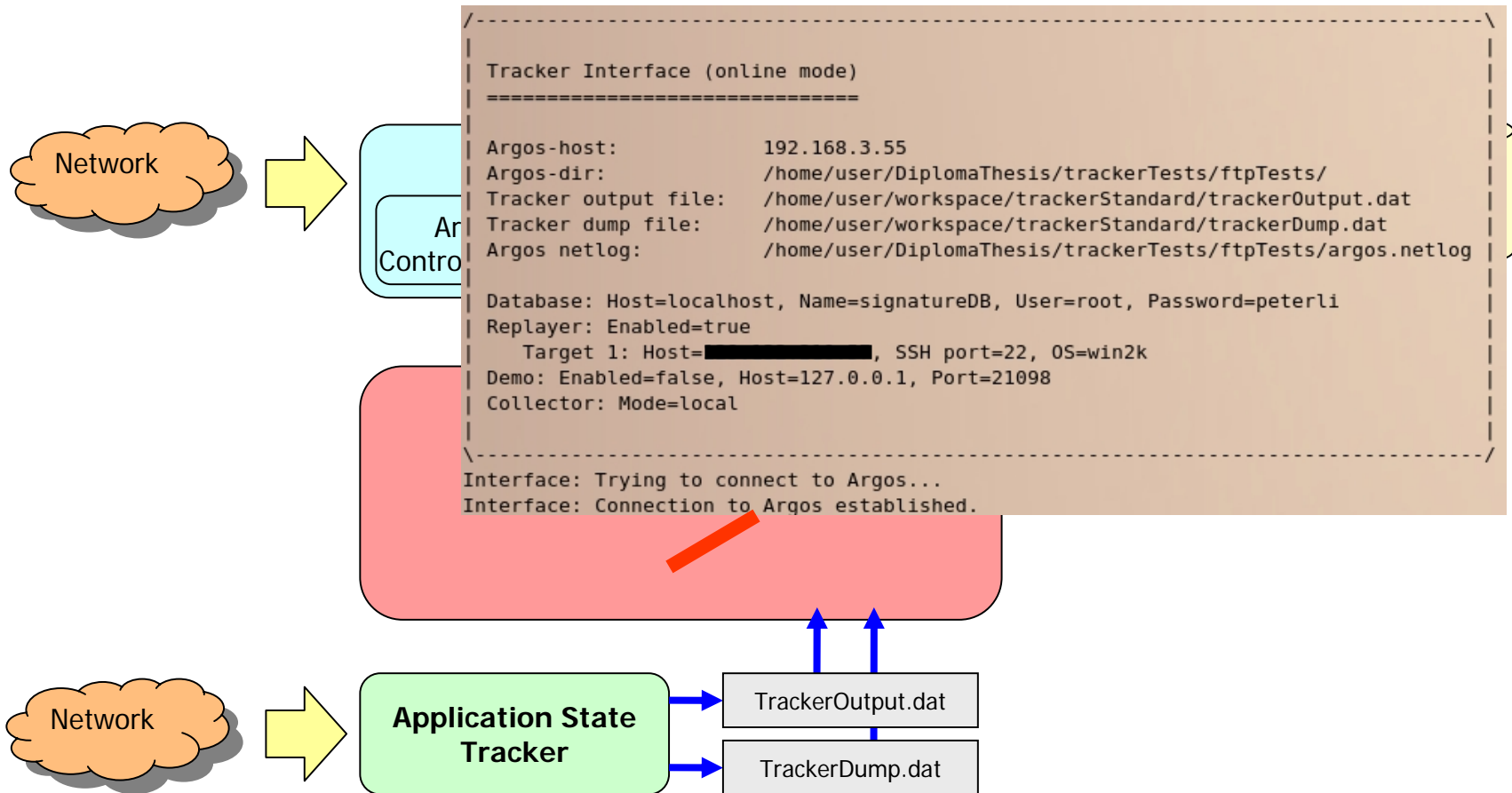
Example: Information extracted by the Analysis Engine



Proof of Concept for Attacks on FTP Services



Proof of Concept for Attacks on FTP Services



Proof of Concept for Attacks on FTP Services

The screenshot displays the ARGOS FTP server control interface. The window title is "ARGOS" and the application is "Idle - WAR-FTPD 1.65". The interface includes a menu bar (Properties, View, Help), a toolbar with various icons, and a main control area. On the right side, there are "System Attributes" and "Messages from the users" sections. The "System Attributes" section includes checkboxes for "Go offline when ready", "Deny all logins (except for administrator)", and "No anonymous logins". It also features input fields for "Max Users" (50) and "Anon." (10), and "IP number and port" (192.168.3.51 and 21). The "Messages from the users" section shows a log of activity, including a user logging out. The status bar at the bottom indicates the server is "ONLINE" with 1 of 32767 sockets, 0 of 50 (16381) users, and 0 file xfers. The taskbar shows the Start button, several icons, and the system tray with the time 8:13 AM.

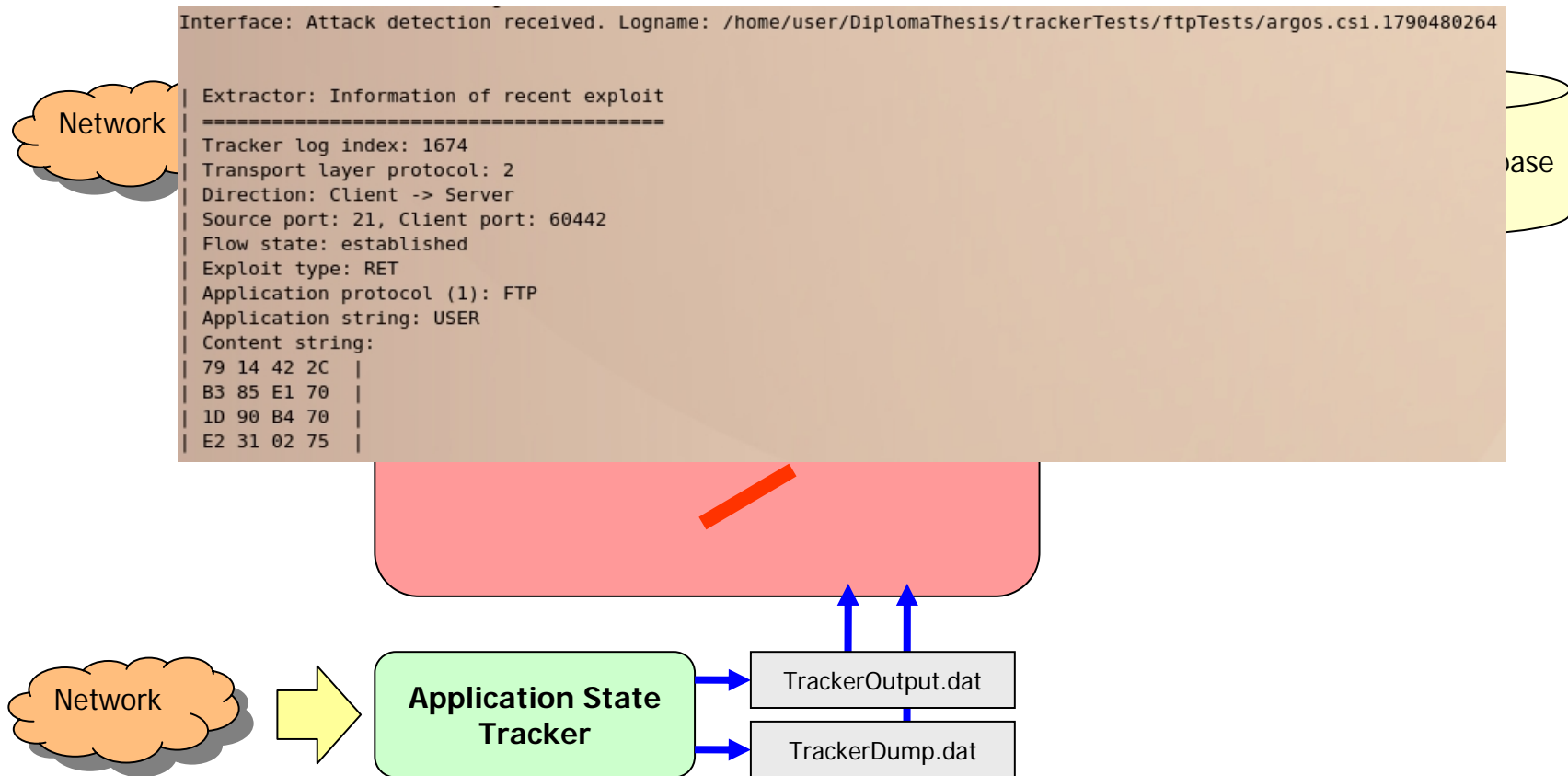
Network

SQL Database

Network

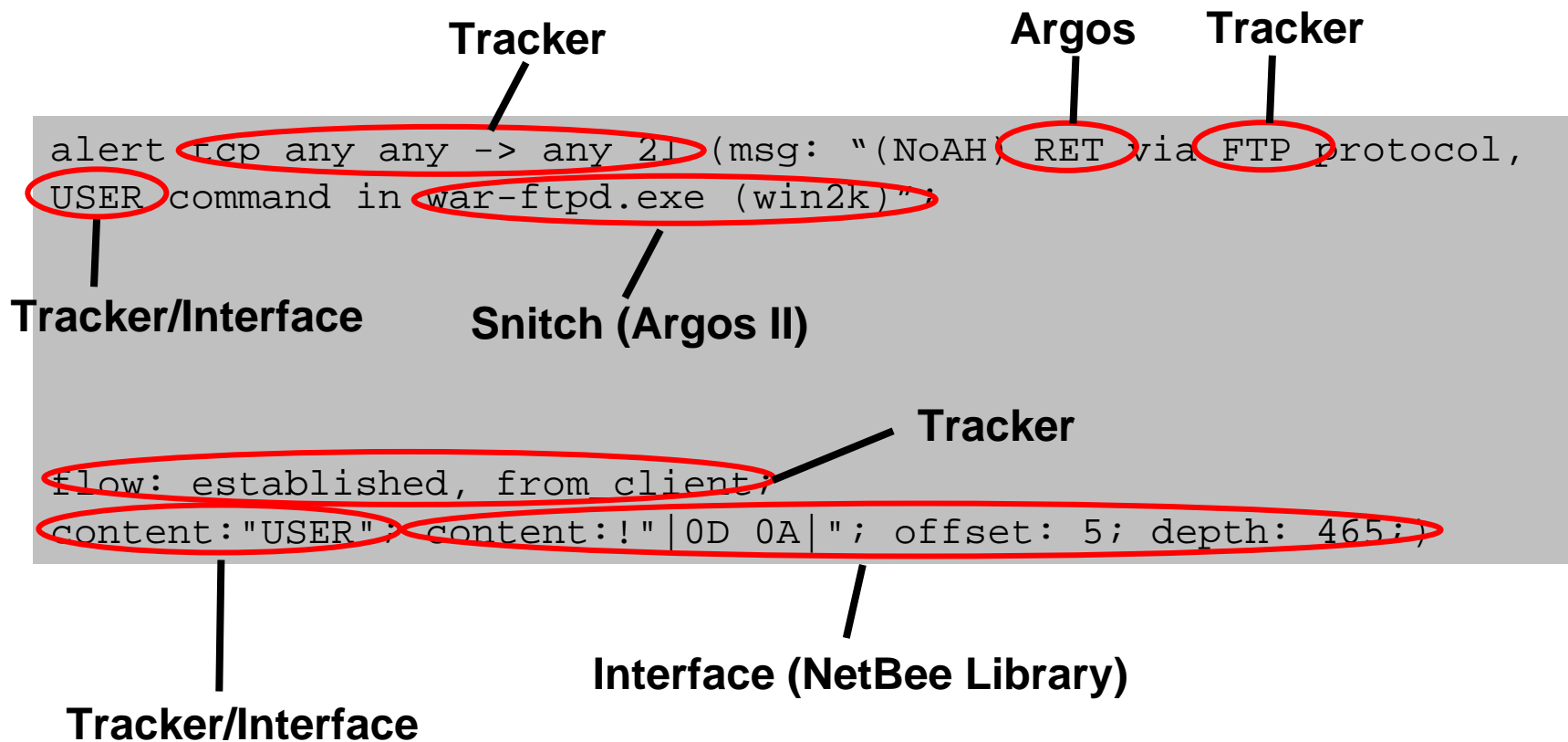
Start | C:\WINNT\system32\cmd... | FTP Idle - WAR-FTPD 1.65 | DE 8:13 AM

Proof of Concept for Attacks on FTP Services



Signature Generation

Generated Signature (WAR-FTPD example):



Results & Conclusion

- Our ASG system generated signatures that have zero false positives
 - Signature describes the vulnerability of the application
 - Signature can be used with SNORT, a well-known IDS
 - Protect server applications from buffer overflows in arbitrary protocols and fields

Generated signatures can compete with other approaches including manually created reference signatures (for attacks within the scope of NoAH)

A blue-tinted photograph of a large, classical-style building with a prominent dome and arched windows, likely a part of the ETH Zurich campus. The image is positioned at the top of the slide.

Questions?

References

- [1] Lam, L. & Chiueh, T.
Automatic Extraction of Accurate Application-Specific Sandboxing Policy
Recent Advances in Intrusion Detection: 7th International Symposium, RAID 2004
- [2] Yegneswaran, Vinod; Giffin, Jonathon T.; Paul Barford & Jha, Somesh
An Architecture for Generating Semantic-Aware Signatures
Proceedings of the 14th USENIX Security Symposium, 2005
- [3] Wang, K. & Stolfo, S.
Anomalous Payload-based Network Intrusion Detection
7th International Symposium on Recent Advances in Intrusion Detection, 2004
- [4] Tang, Y. & Chen, S.
Defending against internet worms: A signature-based approach
Proceedings of IEEE INFOCOM, 2005
- [5] Singh, S.; Estan, C.; Varghese, G. & Savage, S.
Automated Worm Fingerprinting
OSDI, 2004
- [6] Liang, Z. & Sekar, R.
Fast and Automated Generation of Attack Signatures: A Basis for Building Self-Protecting Servers
12th ACM Conference on Computer and Communications Security (CCS), 2005
- [7] Costa, M.; Crowcroft, J.; Castro, M.; Rowstron, A.; Zhou, L.; Zhang, L. & Barham, P.
Vigilante: end-to-end containment of internet worms
Proceedings of the twentieth ACM symposium on Operating systems principles, 2005
- [8] Kim, H. & Karp, B.
Autograph: Toward Automated, Distributed Worm Signature Detection
USENIX Security Symposium, 2004
- [9] Anagnostakis, K. G.; Sidiroglou, S.; Akritidis, P.; Xinidis, K.; Markatos, E. & Keromytis, A. D.
Detecting Targeted Attacks Using Shadow Honey Pots
Proceedings of the 14th USENIX Security Symposium, 2005
- [10] Portokalidis, G.; Slowinska, A. & Bos, H.
Argos: An Emulator for Fingerprinting Zero-Day Attacks
Proc. ACM SIGOPS EUROSYS'2006, 2006
- [11] Newsome, J. & Song, D.
Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software
Proceedings of the 12th Annual Network and Distributed System Security Symposium (NDSS), 2005
- [12] <http://www.nbee.org/>