OPENVMS SECURITY
&
NEW FEATURES IN V8.4

Presenters: Rupesh Shantamurty
OpenVMS Engineering

AGENDA

- Introduction to OpenVMS Security
- New Features in V8.4
- Support for special characters in user names
- HP Code Signing
- Intrusion Detection & SNORT® for OpenVMS
- Q & A

Snort is a registered trademark of Sourcefire, Inc.

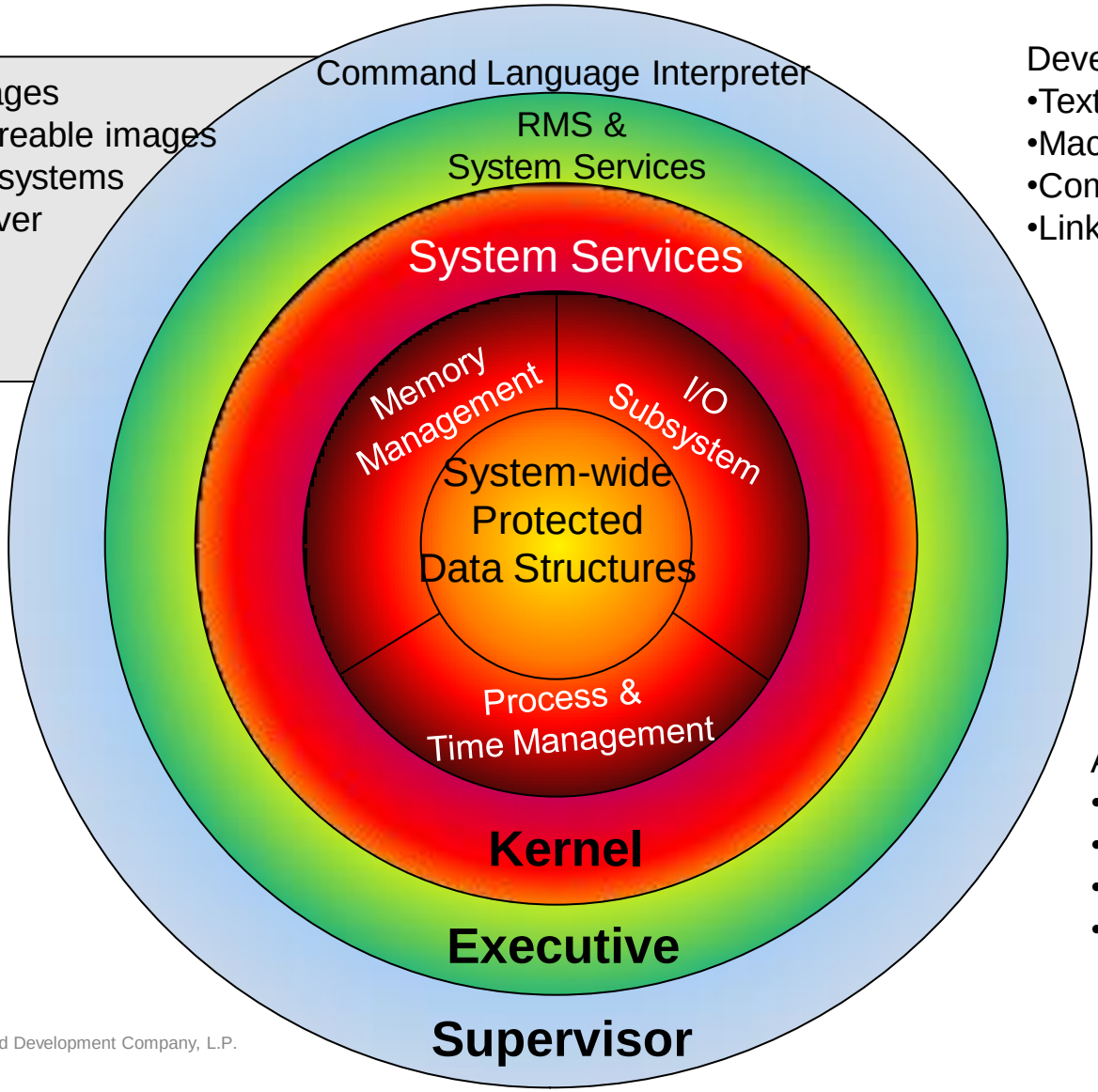
Security = OpenVMS

Introduction to security on OpenVMS

INTRODUCTION

- OpenVMS was designed from day one with the aim of making a “crash proof” system.
- 4 access modes – user / supervisor / exec/ kernel
- Isolates trusted system code from un-trusted user code
- “Firewall” system components to limit the impact of bugs

SYSTEM LAYERING



•Privileged Images
•Protected shareable images
•Protected subsystems
•Privileged server processes

Development Tools
•Text editors
•Macro
•Compilers
•Linker

Run Time Library (General)
•Math library
•String handling
•Screen management
•Misc LIB functions

User

Run Time Library (Language-specific)
•CRTL
•FORTRAN
•PASCAL
•BASIC

Assorted Utilities
•COPY
•HELP
•DIRECTORY
•SORT

SECURITY OFFERINGS

Protecting data: data in transit, data in use, data at rest

Optimal protection against external and internal threats to OpenVMS

Data in transit

- Open SSL
- Kerberos
- Secure Shell
- SFTP/SCP
- Stunnel
- GNUpg

Data in use

- Discretionary access (SOGW)
- ACL's
- Protected subsystems
- Privileged server processes
- Privileged Images
- Protected shareable images

Data at rest

- Encrypt
- Backup/encrypt

Protecting systems: protect-detect-react

Known & unknown vulnerabilities through system minimization & hardening

Protecting identities

Digital Identity life cycle management

Core OS Security

- OS Layering
- Memory protection (KESU/RWE)
- Per thread security (and persona)
- Privilege and Quota checks
- Object access check - Rights identifier etc

Authentication Authorization & Access Control

- SYSUAF – user information/UIC/Privileges/Quotas
- ACME external authentication
- Rights List (roles)
- Enterprise directory
- Third part tools - AAA Server

Protecting against known security defects

- Code signing
- Software Patching (via itrc)

Protecting against unknown vulnerabilities

- Install Time Security (lock down)
- Audit logs
- Host IDS
- SNORT®
- Third part tools for OS hardening

Snort is a registered trademark of Sourcefire, Inc.

NEW FEATURES IN V8.4

NEW FEATURES IN OPENVMS V8.4 (1 of 3)

- Support for special characters in user names – For LDAP based logins
  - Maps user names in domains to user names on OpenVMS system

- New signing and validation mechanism
  - Replaces the existing CDSA based signing & validation mechanism
  - HPBinaryChecker
  - PCSI & VMSINSTAL enhanced to do validation

NEW FEATURES IN OPENVMS V8.4 (2 of 3)

–ACME Enhancements

- ACME LDAP files is now a part of the OpenVMS V8.4 Operating system
  - The Persona extension (Execlet) still needs to be loaded explicitly
- New LDAP configuration Document in SYS$HELP
  - ACMELDAP_STD_CONFIG_INSTALL.PDF
  - ACMELDAP_STD_CONFIG_INSTALL.TXT
- Changes to Upgrade procedure to automatically identify and upgrade the ACME enabled loginout.exe and setp0.exe if present on the system

NEW FEATURES IN OPENVMS V8.4 (3 of 3)

–SSL V1.4

- Based on OpenSSL 0.9.8h
- Vulnerability fixes
- Support for PKCS-12 files
- Support for CMS
- New cipher Camellia
- Visit http://h71000.www7.hp.com/openvms/products/ssl/

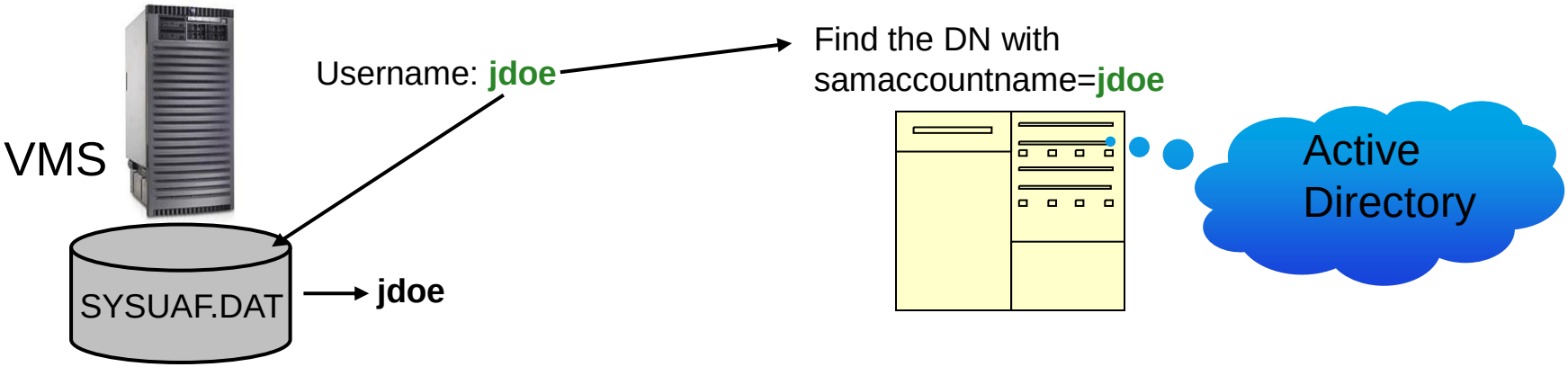
NOTE : SSL V1.4 not backward compatible with SSL V1.3 or earlier versions. SSL V1.3 is based on OpenSSL 0.9.7e stream which is not compatible with 0.9.8 streams.

See Advisory:
http://h20000.www2.hp.com/bizsupport/TechSupport/Document.jsp?lang=en&cc=us&objectID=c02449766

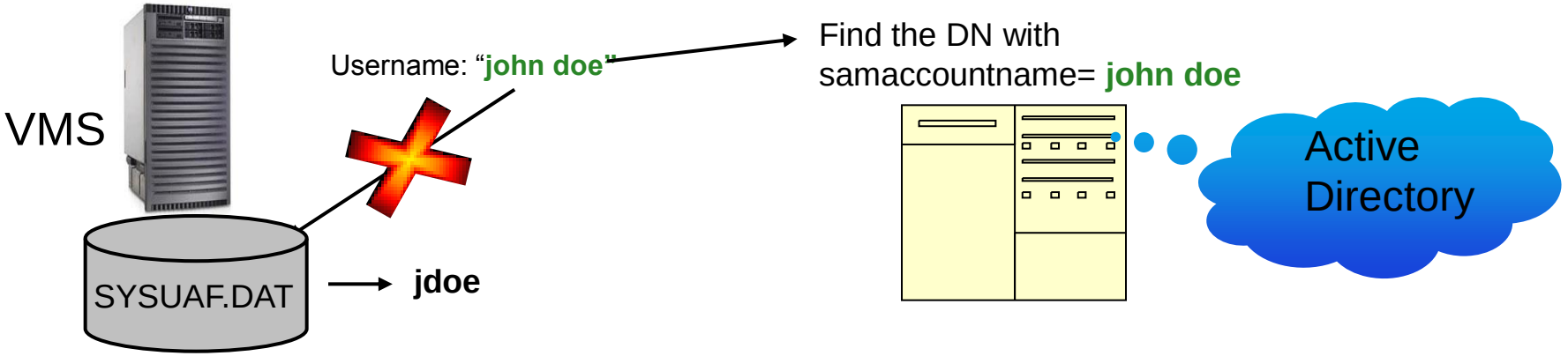
V8.4 Feature – Special characters in Usernames

Requirement for Special characters in User Names

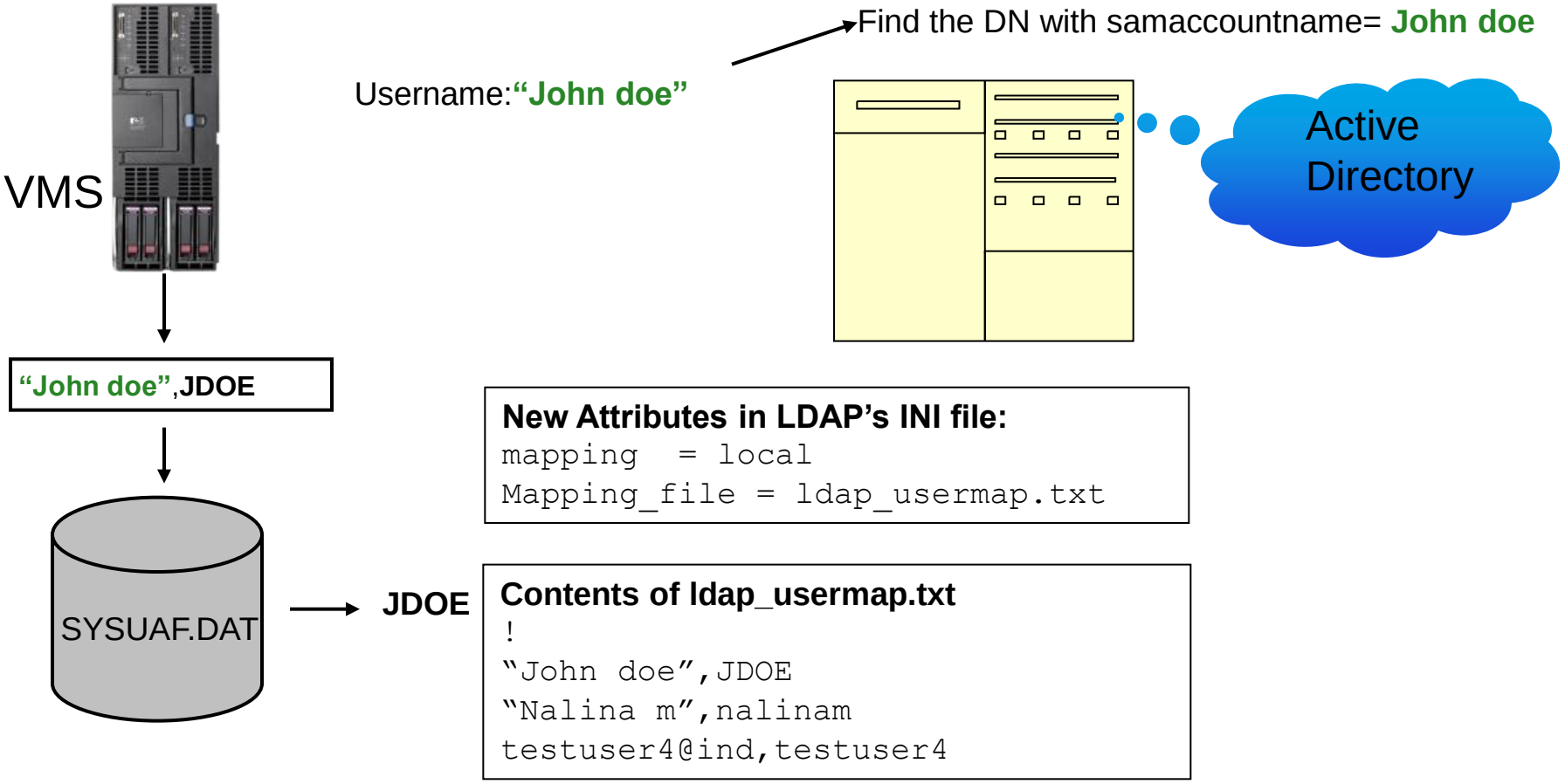
Before OpenVMS V8.4 : one-to-one mapping



"John doe" ≠ jdoe



LOCAL MAPPING SOLUTION IN V8.4



- Use the LDAP_LOAD_LOCALUSER_DATABASE.EXE to load new users without stopping/restarting the ACME_SERVER

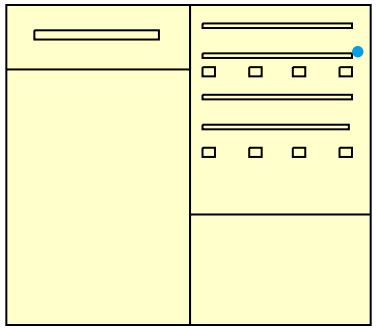
```
$ ldap_loaddb:=="$SYS$COMMON:[SYSEXE]LDAP_LOAD_LOCALUSER_DATABASE.EXE"
$ ldap_loaddb SYS$COMMON:[SYS$STARTUP]LDAP_LOCALUSER_DATABASE.TXT
```

GLOBAL MAPPING SOLUTION IN V8.4

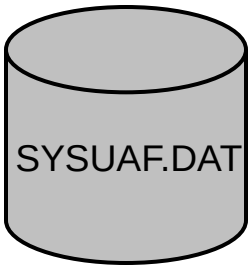
VMS

Username: **“John doe”**

Find the DN with
samaccountname=**“John doe”**



Active Directory



JDOE

DN: cn=john doe,…
samaccountname: **John doe**
description: **abcd/JDOE**

Attributes in LDAP’s INI file :

```
mapping = server
mapping_attribute = description
mapping_target = abcd
```

HP CODE SIGNING

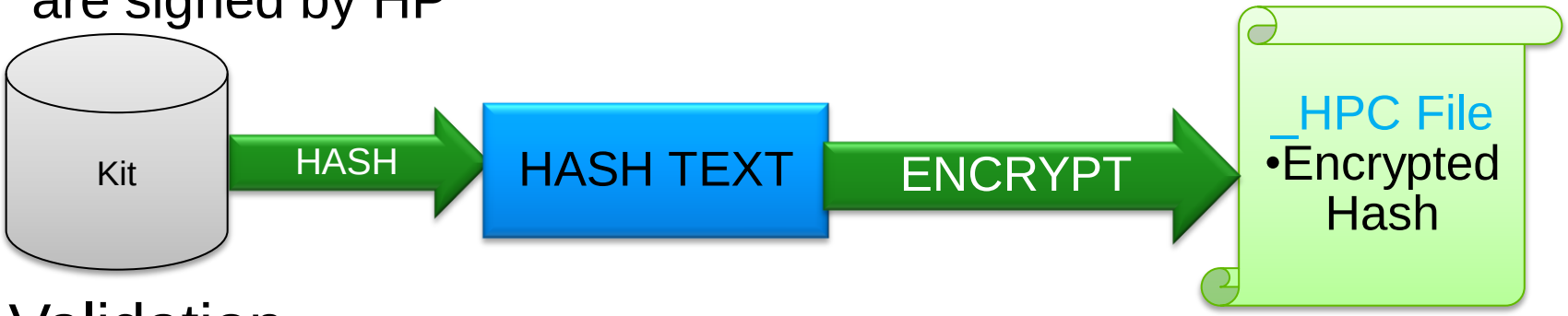
Why do customers need Code Signing ?

– When HP signs code, our customers can answer two specific questions:

- Did this code come from HP ? (**authenticity**)
- Has this code been altered since HP signed it ? (**integrity**)

Signing & Validation on OpenVMS V8.4

– Signing
  - All kits, including the OpenVMS I64 V8.4 operating system kit, are signed by HP



– Validation
  - New Product HPBINARYCHECKER is included along with OpenVMS V8.4 to enable the validation of the signature (manifest).
  - PCSI & VMSINSTAL use the HPBINARYCHECKER to validate the signature. This validation would happen at the beginning of the installation of the kit.

COMPATIBILITY

- PCSI kits that were generated Pre-V8.4 with the CDSA signatures(_ESW) will get validated and installed on V8.4
- PCSI kits without any signature file will not be validated and the option of installation without validation is given to the user.
- Kits generated Pre-V8.4 can also be installed on V8.4
- Kits with new signatures(_HPC) will not be validated on Pre-V8.4

MANDATORY SOFTWARE SSL V1.4

– HPBINARYCHECKER uses SSL V1.4 product.

Do not remove the SSL product.

- In case SSL product is removed. It has to be reinstalled before any other validation can occur. The SSL PCSI kit itself can be installed by the below approaches:
  - deleting the _HPC file (manifest for HPBINARYCHECKER). The PCSI will use the _ESW file (Manifest for CDSA validation)
  - Installing the SSL PCSI kit with /OPTIONS=NOVALIDATE_KIT qualifier

VISIBLE CHANGES

- Every signed kit has a signature file with the name as <full kit name>_HPC.
  - e.g.

```
HP-VMS-AVAIL_MAN_COL-V0301--1.PCSI$COMPRESSED;6
HP-VMS-AVAIL_MAN_COL-V0301--1.PCSI$COMPRESSED_ESW;3
HP-VMS-AVAIL_MAN_COL-V0301--1.PCSI$COMPRESSED_HPC;3
```

PCSI KIT INSTALLATION

– The installation logs will now show "HPCVALPASSED" instead of just "VALPASSED"

```
$ prod instal *

Performing product kit validation of signed
 kits ...

%PCSI-I-HPCVALPASSED, validation of
 MYHAPY$DKA0:[DEMO]HP-VMS-AVAIL_MAN_COL-
 V0301--1.PCSI$COMPRESSED;6 succeeded
```

VMSINSTAL KIT INSTALLATION

- Validation statement will appear as given below

```
************************************************************

%VMSINSTAL-I-VALSIGN, Performing product kit validation of signed kits ...

%VMSINSTAL-I-VALPASSED, validation of MORGAN$DKA100:[DEMO]CMS452.A_HPC succeeded

%VMSINSTAL-I-VALPASSED, validation of MORGAN$DKA100:[DEMO]CMS452.B_HPC succeeded

************************************************************
```

- The history file look as give below with a new column for validation

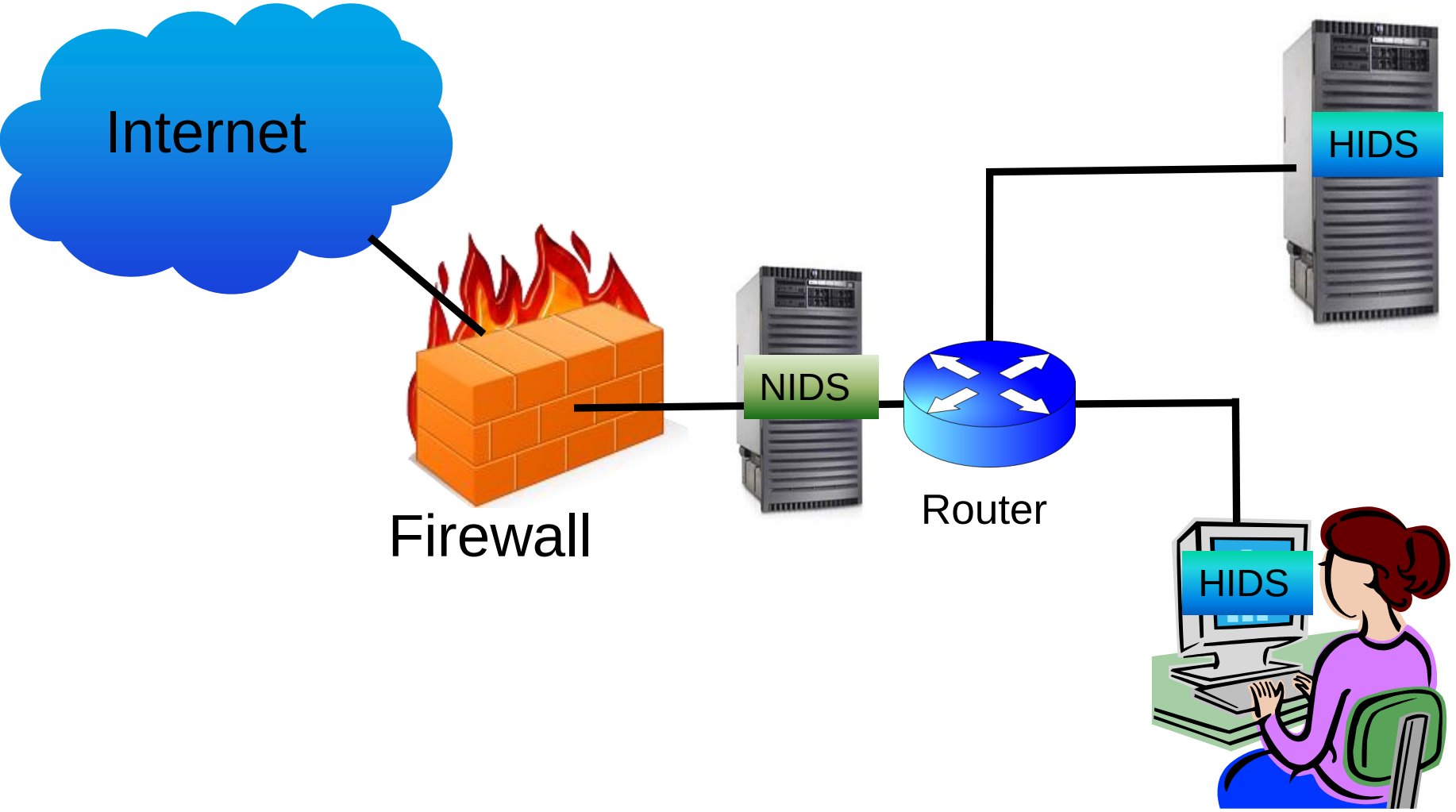
```
$ type sys$update:vmsinstal.history

VMSINSTAL Product Installation History File

S = Success    F = Failure    NA = Not Applicable    U = Unsigned Product    HPC = Signed Product

-------------------------------------------------------------------------------------------------------------

Product Information                                                  | Installation Information

---------------------------------------------------------------------|---------------------------------------

Name                     | Mnemonic         | Version | Date        | Status| IVP   | Node   | Installer    | VALSIGN

---------------------------------------------------------------------|---------------------------------------

                         | CMS              | 45.2    |  3-SEP-2010 | S     | S     | MORGAN | SYSTEM       | HPC
```

SNORT® FOR OPENVMS

Intrusion Detection System

Snort is a registered trademark of Sourcefire, Inc.

INTRUSION DETECTION SYSTEM



Types of IDS based on mode of operation

- Signature-based
  - Intruders have signatures (computer viruses).
  - Compare packets against a database of signatures or rules from known malicious threats.
  - Log suspicious activity and generate alerts.
- Anomaly-based
  - Depends on packet anomalies present in protocol header parts.
  - Compare network traffic against an established normal network traffic evaluated baseline

INTRODUCTION TO SNORT®

- SNORT® is primarily a rule-based IDS
  - Uses a rule-driven language
- input plug-ins to detect anomalies in protocol headers.
- Combines the benefits of signature, protocol and anomaly based inspection methods.
- Open source network intrusion detection and prevention system (IDS/IPS)
- Developed by Sourcefire
- Most widely deployed IDS/IPS technology worldwide

Snort is a registered trademark of Sourcefire, Inc.

INSTALLATION & SETUP

Software requirements/Prerequisites to run SNORT®

- Download Kit from
  - http://h71000.www7.hp.com/openvms/products/snort/
- Operating System/Architecture:
  - HP IA64VMS OPENVMS V8.3-1H1 onwards
- Other Products:
  - HP I64VMS SSL V1.4-335
  - HP I64VMS TCPIP V5.6-9ECO5 or later
  - JFP I64VMS MYSQL051 V22.0-0 or later (**If MySQL logging is required**)
  - JFP I64VMS ZLIB V1.2-3 or later
- Disk:
  - **ODS-5 disk**

Snort is a registered trademark of Sourcefire, Inc.

USING SNORT® ON OPENVMS

MODES OF SNORT®

- Sniffer mode

```
./snort -v
```

  - Reads the packets off of the network

- Packet Logger mode

```
./snort -dev -l ./log
```

  - Logs the packets to disk.

- Network Intrusion Detection System Mode

```
./snort -v -l ./log -c
snort.conf
```

  - Apply the rules configured in the snort.conf

- **Inline Mode (*)**

```
./snort -Qc
../etc/drop.conf
```

  - Packets from iptables instead of libpcap
  - Capability to drop packets

*** Currently not supported in OpenVMS**

OUTPUT LOG FILES

- Logging can be done in binary, ascii and unified binary formats.
- Default logging directory :
  `SNORT$SPECIFIC:[000000.VAR.LOG.SNORT]`
- To override the logging directory
  - use –l runtime argument "`$snort –v –l ./log`"
  - Redefine the `SNORT$SPECIFIC` logical to a different directory in `SNORT$LOGICALS.COM`
- **Alert file:**
  `SNORT$SPECIFIC:[000000.VAR.LOG.SNORT]alert.;1`
- **Log file:**
  `SNORT$SPECIFIC:[000000.VAR.LOG.SNORT]SNORT.LOG.1280890416;`
- Syslog output
  - Snort on OpenVMS provides wrapper for Syslog which logs all Snort alerts and messages into `snort$specific:[var.log.snort]syslog.log`

Demo - Snort Attack Stimulation

Q&A