

VELOCITY  
SOFTWARE

## *VM and SSL*

Velocity Software Inc.  
196-D Castro Street  
Mountain View CA 94041  
650-964-8867

Velocity Software GmbH  
Max-Joseph-Str. 5  
D-68167 Mannheim  
Germany  
+49 (0)621 373844

**Rick Troth**  
**Velocity Software**  
[<rickt@velocitysoftware.com>](mailto:rickt@velocitysoftware.com)  
<http://www.velocitysoftware.com/>

VM and Linux Workshop 2012  
University of Kentucky

Copyright © 2012 Velocity Software, Inc. All Rights Reserved. Other products and company names mentioned herein may be trademarks of their respective owners.

# Disclaimer

The content of this presentation is informational only and is not intended to be an endorsement by Velocity Software. (ie: I am speaking only for myself.) The reader or attendee is responsible for his/her own use of the concepts and examples presented herein.

In other words: Your mileage may vary. “It Depends.”  
Results not typical. Actual mileage will probably be less.  
Use only as directed. Do not fold, spindle, or mutilate. Not to be taken on an empty stomach. Refrigerate after opening.

In all cases, *“If you can't measure it, I'm just not interested.”*

Foundations of SSL

Authenticating the other party

Securing the session or transaction

Overview using zSSL

Overview using VM SSL

Related topics: SSH, PGP/GPG

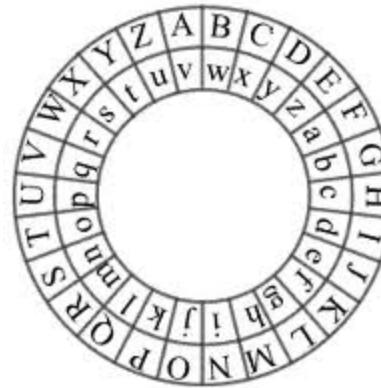
# SSL – the history of encryption

## Early ciphers

- Caesar
- Jefferson
- Enigma, Lorenz

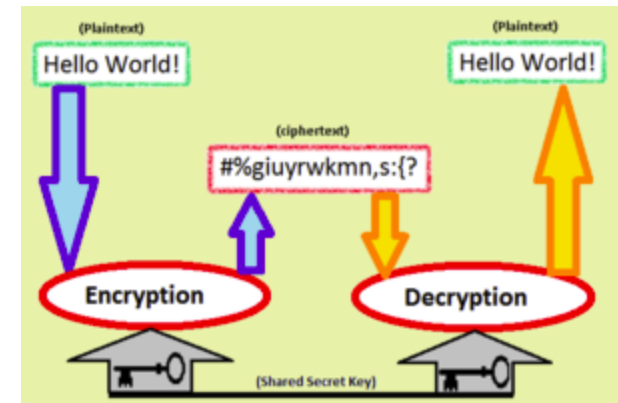
## Passwords

## One-time use



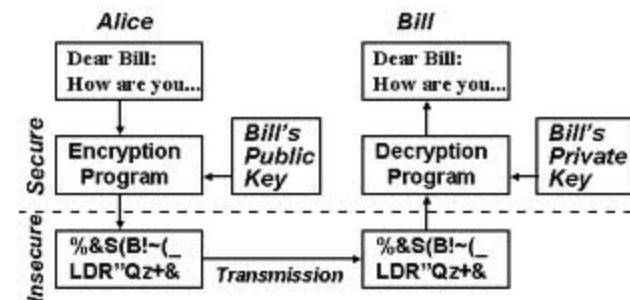
# Asymmetric Crypto

What if someone  
got the password?



Rivest, Shamir, Adleman  
involves a public key and a private key  
AKA: asymmetric

[http://en.wikipedia.org/wiki/  
Public-key\\_cryptography](http://en.wikipedia.org/wiki/Public-key_cryptography)



# *Encryption plus Authentication*

Encrypt with public key (of recipient)

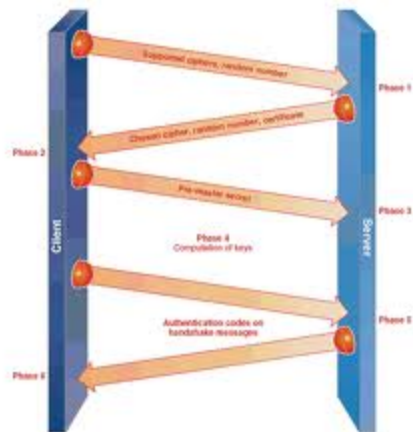
Decrypt with secret key

Sign with secret key

Verify with public key (of sender)

# Don't Talk to Strangers

Authenticate the server  
Establish a secure channel  
Uses existing network



Message Types	
Code	Description
0	HelloRequest
1	ClientHello
2	ServerHello
11	Certificate
12	ServerKeyExchange
13	CertificateRequest
14	ServerHelloDone
15	CertificateVerify
16	ClientKeyExchange
20	Finished

Does not protect “data at rest”

# Transport Layer Security

Handshake authenticates  
SSL provides a “channel”  
Compare to SSH  
Contrast with PGP/GPG (data at rest)

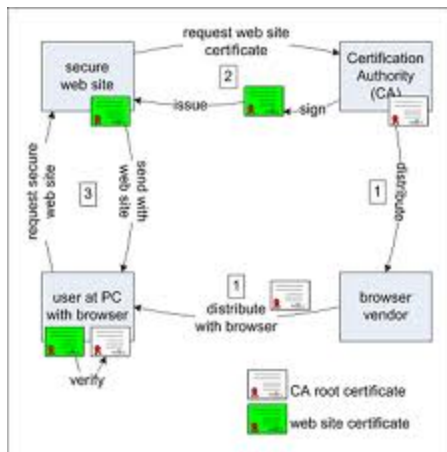
Content types

Hex	Dec	Type
0x14	20	ChangeCipherSpec
0x15	21	Alert
0x16	22	Handshake
0x17	23	Application

+	Byte +0	Byte +1	Byte +2	Byte +3
Byte 0	Content type			
Bytes 1..4	Version		Length	
	(Major)	(Minor)	(bits 15..8)	(bits 7..0)
Bytes 5..(m-1)	Protocol message(s)			
Bytes m..(p-1)	MAC (optional)			
Bytes p..(q-1)	Padding (block ciphers only)			



# Public Key Infrastructure



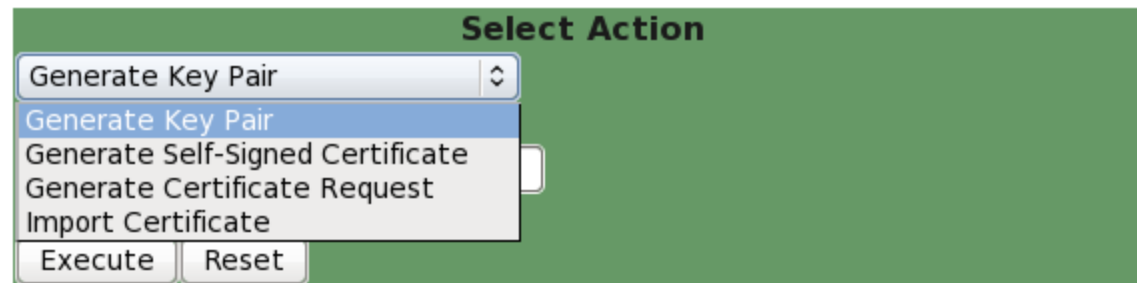
CA certificate pre-loaded  
WS requests assertion  
CA signs WS request  
WS loads that  
Browser hits WS,  
compares signature chain  
Browser/WS agree on  
session keys



# *zSSL Administration Functions*

Generate a key pair and a self-signed cert

`http://vmsys.you.tld:81/portal/dosssladm.cgi`



The screenshot shows a web interface with a green background. At the top, the text "Select Action" is displayed. Below it is a dropdown menu with the following options: "Generate Key Pair", "Generate Key Pair", "Generate Self-Signed Certificate", "Generate Certificate Request", and "Import Certificate". The first two options are identical, and the second one is highlighted with a blue background. Below the dropdown menu are two buttons: "Execute" and "Reset".

# *zSSL Administration Functions*

Select “Generate Key Pair”

Enter “mycert” for key name (a CMS FN)

choose key size – 2048 bits takes half a min

“2048 bit key 'MYCERT' created.”

Select “Generate Self-Signed Certificate”

Fill-in the blanks, then click “Execute”

“Certificate 'MYCERT' created.”

# zSSL Self Signed Certificate

**Signer Key Name:**  
-same-as-certificate- ↕

Enter Common Name (required, usually a TCP/IP Domain Name System host name):

Enter Locality/City:

Enter Organization Name (required, usually a legal name under which organization is registered):

Enter Province/State:

Enter Organization Unit (some division in the organization):

Enter two-character ISO Country code:

Enter Email Address (contact email address):

Enter Serial number (1-999999):

Enter Period (1-99 days):

## *Copy (Move) zSSL key and cert*

```
vmklink .dir sfszvps:zadmin. ( w fi
```

Copy MYCERT KEYP and MYCERT X509CERT  
from ZADMIN “A” to config directory “C”.

Delete the originals from ZADMIN “A”. (maybe not  
right away)

# Copy (Move) zSSL key and cert

```
File Options
TROTHR FILELIST A0 V 169 Trunc=169 Size=4 Line=1 Col=1 Alt=0
Directory = SFSZVPS:ZADMIN.
Cmd Filename Filetype Fm Format Lrecl Records Blocks Date Time
- MYCERT X509CERT Z1 V 894 1 1 2012-05-21 14:25:25
MYCERT KEYP Z1 V 1191 4 1 2012-05-21 14:17:34
LASTING GLOBALV Z1 V 35 3 1 2012-05-07 13:31:40
PROFILE EXEC Z2 V 62 97 1 2011-10-04 14:29:16

1= Help 2= Refresh 3= Quit 4= Cancel 5= Sort(dir) 6= Sort(size)
7= Backward 8= Forward 9= FL /n 10= Share 11= XEDIT/LIST 12= Cursor
====>

X E D I T 1 File
004/001
```

# CONFIG ZWEBS1

Create CONFIG ZWEBS1 thru ZWEBS5  
(or however many you want) as well as server  
virtual machines similar to ZWEBnn

Modify PORT statement  
“PORT 443 MYCERT”

Authorize ZWEBSn for TCP port 443  
(but do not use VM SSL for these)

Remember ZWEBS0 for “admin” function

Start em up!

# Server with Self-Signed Cert



## This Connection is Untrusted

You have asked Firefox to connect securely to **192.168.5.44:2983**, but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

### What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

Get me out of here!

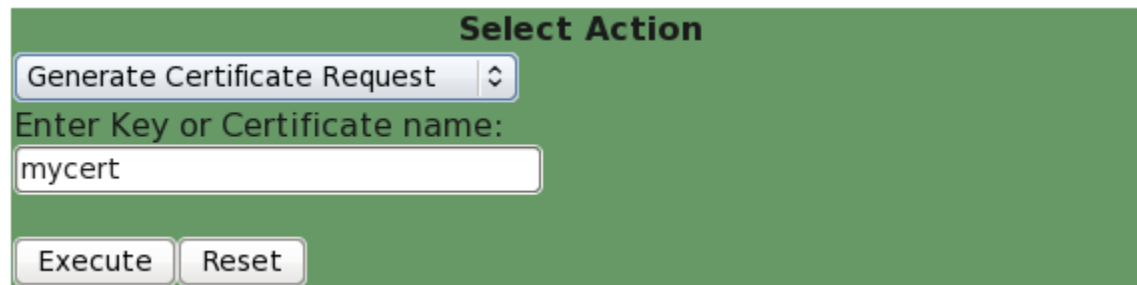
- ▶ **Technical Details**
- ▶ **I Understand the Risks**



# Generate a “Certificate Request”

Still got that key file? (revise prior steps)

Before you delete key file from ZADMIN “A”



Select Action

Generate Certificate Request

Enter Key or Certificate name:

mycert

Execute Reset

# Generate a “Certificate Request”

Same X.509 data as for self-signed ...

Enter Common Name (required, usually a TCP/IP Domain Name System host name):	Enter Locality/City:
<input type="text" value="rmtzvm.velocitysoftware.com"/>	<input type="text" value="Grove City"/>
Enter Organization Name (required, usually a legal name under which organization is registered):	Enter Province/State:
<input type="text" value="Velocity Software"/>	<input type="text" value="OH"/>
Enter Organization Unit (some division in the organization):	Enter two-character ISO Country code:
<input type="text" value="TrothR"/>	<input type="text" value="US"/>
Enter Email Address (contact email address):	
<input type="text" value="rickt@velocitysoftware.com"/>	
<input type="button" value="Execute"/>	<input type="button" value="Reset"/>

# Generate a “Certificate Request”

Copy-n-paste PEM format cert request ...

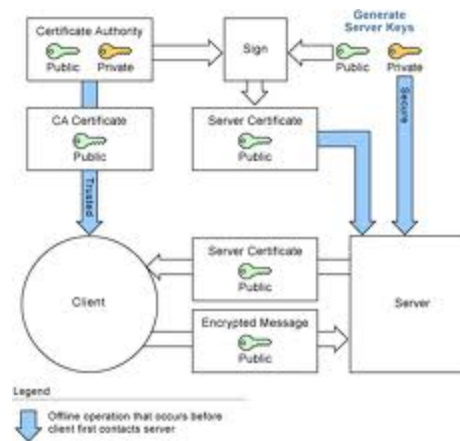
**Copy and Paste this Certificate Signing Request according to your Certificate Authority interface.**

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIC8TCCAdkCAQAwga0xCzAJBgNVBAYTAlVTMQswCQYDVQQIDAJPSDEMBEGA1UE
BwwKR3JvdmlUgQ2l0eTEaMBGGA1UECgwRVmVsb2NpdHkgU29mdHdhcmUxDzANBgNV
BAAsMBIyRyb3RoUjEKMCIgA1UEAwbcm10enZtLnZlbG9jaXR5c29mdHdhcmUuY29t
MSkwJwYJKoZIhvcNAQkBFhpyaWNrdEB2ZwvY2l0eXNvZnR3YXJLLmNvbTCCASIw
DQYJKoZIhvcNAQEBBQAdggEPADCCAQoCggEBAKyG44B6tpDpshXj52TuW+q/zjI
UeaRSq/XQfSS39P6d9nC2gSfKkm5OwYcLZ8v0eRHovgEFyy8Uoxk076MlI0ySNx6
vk7SLwhci6Bmymo7u/7tDwxPPz6Pq4T/SrkW/tqhtHusJgm+/hXrqvvW8Nbm5Wu7
hyCxnG03iRfQ/H29ECILu0xCb795mm1AUpEsZYHAa/hMdAf/wGDbIxmPxSCKVT2
KbHxcQQHuyvLlriBQJ5r/BwB1qa6CisgDAwpqX6U16MXZmxpr00YC6/GGN/A96n
8Ndyp6aWx5kibMdxLdfHmyoGuczgTjRQ42eS0SwwjRkk77Ed4ngH9+z5gNUCAwEA
ATANBgkqhkiG9w0BAQQFAAOCAQEA8xeyavOeCPkHk0xNPj76AS6ux8Yxavirb/S
svmb8oyZI+dyHADxqPLJf/jKgh4LVPqw0GLV8mDs3w2nGlqMQTAe0mnl05+EdZyM
q7J0Yy8ENJS3YnTJkvaTIRCuwW75JrNZGtSkRNyots4D149V0qbQviHKRYGg9d0Q
KhyFTdCcxbUS6y7b0iupoesMTRYuj/TZVs4v6US1qldAsiKp+b7Z8djI8ONPTU8
5++tTVluvScVYAeuluEbF6u1kuyLtcnP4FpIudrAYBmbG9g/G3pKjwBYexpIrL30
Sb666RbqccvKRjPmX1Ndndm3VVQvm1p/NpafsJJwGu0q/+MkAg==
-----END NEW CERTIFICATE REQUEST-----
```

... and send it to your CA.

## VM SSL Key Management

## Set up GSKADMIN and wire it into the stack



# Sign onto GSKADMIN

## Use 'gskkyman' command

# VM SSL Key Management

```
File  Options
Ready; T=0.01/0.01 12:51:46
gskkyman

Database Menu

1 - Create new database
2 - Open database
3 - Change database password
4 - Change database record length
5 - Delete database
6 - Create key parameter file
7 - Display certificate file (Binary or Base64 ASN.1 DER)

0 - Exit program
Enter option number:
```

—

RUNNING ZVMV5R40

031/001

# VM SSL Key Management

## Create a key database ...

- Option 1
- Filename “Database.kdb”
- 3700 days = 10 years, 6 weeks
- Default record size

## Fix file access ...

```
openvm permit /etc/gskadm/Database.kdb rw- r-- ---  
openvm permit /etc/gskadm/Database.sth rw- r-- ---
```

# VM SSL Key Management

```
File  Options

Key Management Menu

Database: /etc/gskadm/Database.kdb
Expiration: 2022/07/30 21:38:58

1 - Manage keys and certificates
2 - Manage certificates
3 - Manage certificate requests
4 - Create new certificate request
5 - Receive requested certificate or a renewal certificate
6 - Create a self-signed certificate
7 - Import a certificate
8 - Import a certificate and a private key
9 - Show the default key
10 - Store database password
11 - Show database record length

0 - Exit program

Enter option number (press ENTER to return to previous menu):

-

RUNNING  ZVMV5R40
031/001
```

# VM SSL Key Management

## Create a self-signed certificate ...

- Option 6
- Option 7, server cert with 4096-bit RSA key
- Option 3, SHA-256 signature digest
- Enter a label, UPPER CASE
- Enter X.509 stuff

Apply that label to a “secured” TCP port



# VM SSL Key Management

## Create new certificate request ...

- Option 4
- Option 3, cert with 4096-bit RSA key
- Enter filename
- Enter a label, UPPER CASE again
- Enter X.509 stuff

File is PEM encoded; send to your CA

## Peer-to-Peer

- PGP style

## Third Party / Centralized

- PKI style

## Manual Assertion

- Self-signed certs

Question: which works best for business?

## SSL and TLS (PKI)

- originally for HTTPS, now others too
- third party trust
- X.509 certificates

## SSH

- variable trust models
- keys

## PGP/GPG

- peer-to-peer trust
- keys

## 'ssh-keygen' command

- Generates pub / sec (no “.pub”), two files

Send pub to “authorized\_keys” file  
of target user(s) on target system(s)

## Generate a key pair

```
gpg --gen-key
```

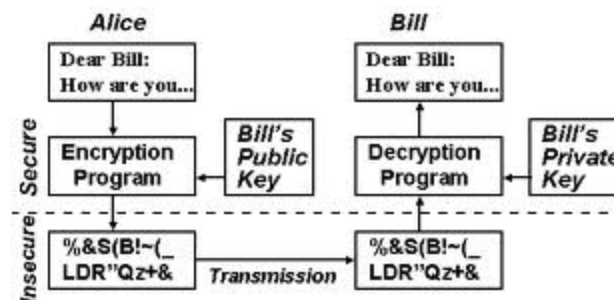
## Export your pub key, sign others

```
gpg --armor --export
```

```
gpg --sign-key user-id
```

## Import signed keys and signatures

```
gpg --import
```



# *Terms and Tools to Learn*

Certificates identified by SDN,  
“subject distinguished name”

X.509 verbiage abounds

Need overview of BFS files (for VM SSL)

- x /etc/gskadm/mycert.crq (nam bfs

# *What is a “subject”?*

## What is the “subject”?

- That which is “signed” by an “authority”

## What is the “authority”?

- That which cryptographically signs the “subject”

maximum entropy, minimum energy

maximum entropy, minimum “order”

Entropy ==> Randomness

Strong encryption

requires reliable randomness



# *Water Cooler Leaks*

## Human factors remain the biggest risk

- Easy passwords
- Gullible to scams
- Profiled for info
- Unsecured hardware
- Lost hardware



## DNS-Based Authentication of Named Entities

Alternative to traditional CAs  
Requires DNSSEC

You need SSL

Apply SSL carefully

Understand the concepts

Be prepared:

SSL is a moving target!