POC2016 - Flip Feng Shui: Hammering a Needle in the Software Stack

Kaveh Razavi Ben Gras Erik Bosman
Bart Preneel Cristiano Giuffrida Herbert Bos

November 10, 2016

VU UNIVERSITY AMSTERDAM

Who am I

- Security researcher in academia
- VU University in Amsterdam, systems security research group (vusec)
- Shown left: Kaveh and Ben after submitting this work to Usenix Security

Who are we

- Shown left: The rest of the vusec group at the VU
- We publish offensive and defensive systems security research at security conferences
- Also software reliability research

Teaser

- OpenSSH compromise
- apt-get compromise by GPG signature forgery
- No software bug
- Weak assumptions
- Demo!

Contribution

Flip Feng Shui is a novel exploitation structure

- Hardware glitch
- Memory massaging primitive

Makes the glitch

- Easy to target precisely
- Reliable

We demonstrate FFS $=$ Rowhammer $+$ Memory Deduplication

Outline

Flip Feng Shui At Work

Outline

Flip Feng Shui At Work

Flip Feng Shui Mechanics

Outline

Flip Feng Shui At Work

Flip Feng Shui Mechanics

OpenSSH Attack

Outline

Flip Feng Shui At Work

Flip Feng Shui Mechanics

OpenSSH Attack

Privilege Escalation Bitflips

Outline

Flip Feng Shui At Work

Flip Feng Shui Mechanics

OpenSSH Attack

Privilege Escalation Bitflips

GPG/APT Updates Attack Demo

Outline

Flip Feng Shui At Work

Flip Feng Shui Mechanics

OpenSSH Attack

Privilege Escalation Bitflips

GPG/APT Updates Attack Demo

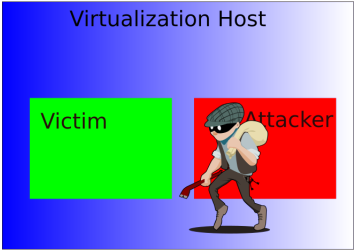
Notification, Conclusion & Further Resources

Section 1

Flip Feng Shui At Work

Flip Feng Shui

- Flip one bit per page in a co-hosted victim VM



- Whenever you know its contents
- Organised bitflip
- DRAM glitch
- Breaks CPU virtualization isolation

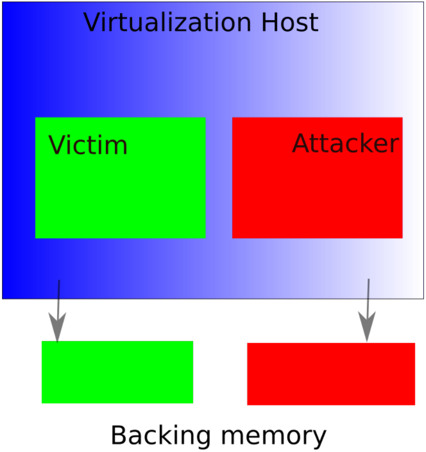
Section 2

Flip Feng Shui Mechanics

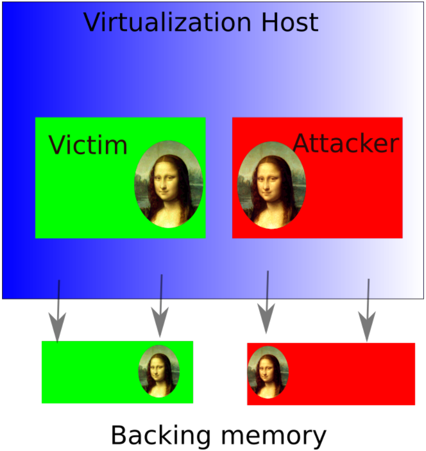
Flip Feng Shui Mechanics

- Co-hosted VMs
- Memory deduplication
- Rowhammer
- RSA

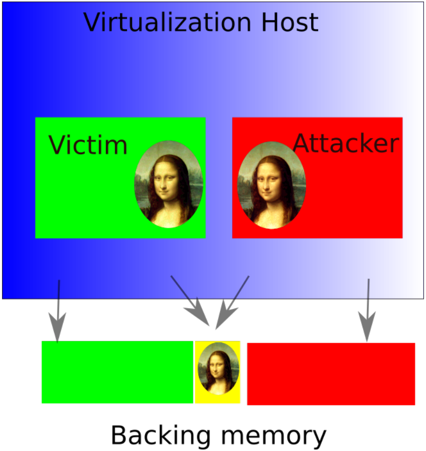
Memory deduplication



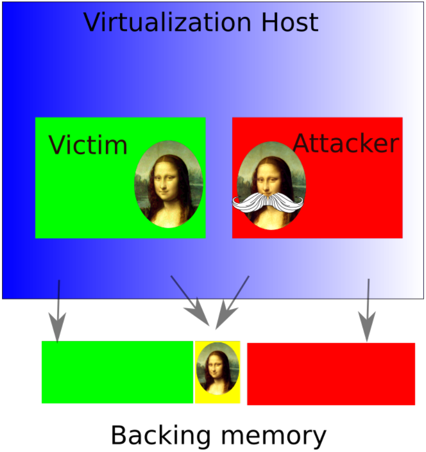
Memory deduplication



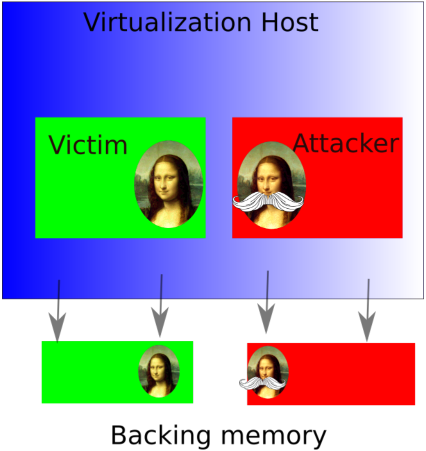
Memory deduplication



Memory deduplication

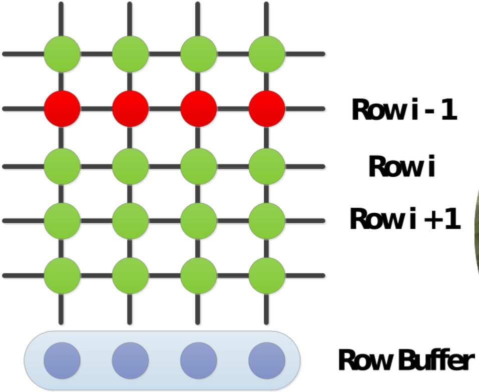


Memory deduplication



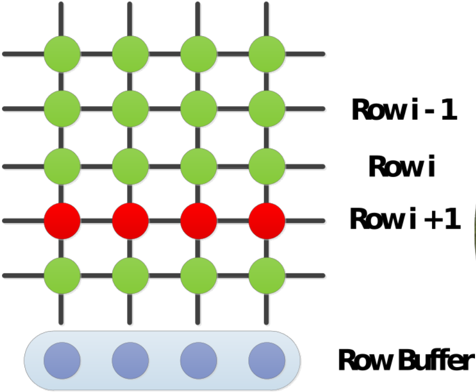
Rowhammer

- Causes charge to leak in DRAM
- DRAM row activations cause flips



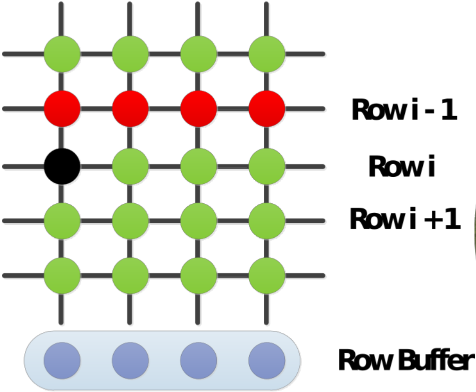
Rowhammer

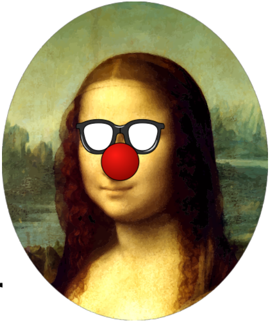
- Causes charge to leak in DRAM
- DRAM row activations cause flips



Rowhammer

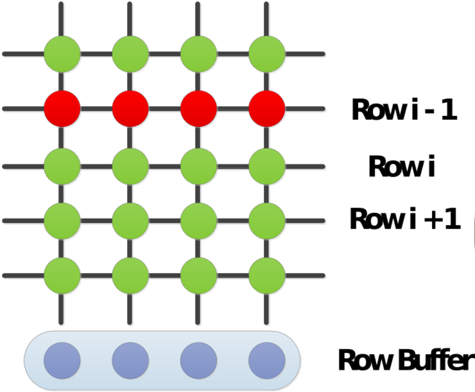
- Causes charge to leak in DRAM
- DRAM row activations cause flips





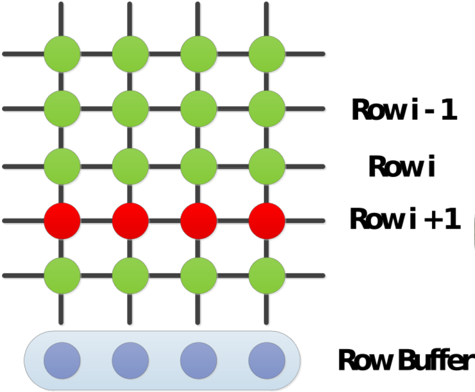
Rowhammer

- Causes charge to leak in DRAM
- DRAM row activations cause flips



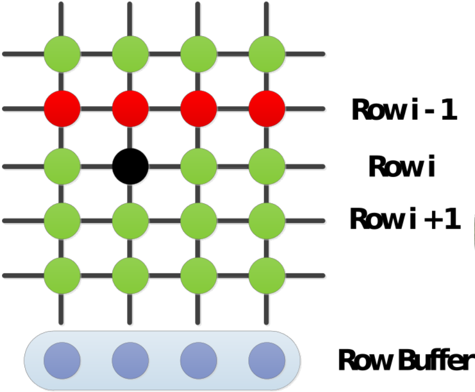
Rowhammer

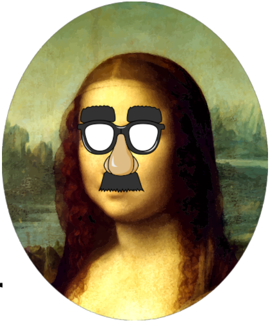
- Causes charge to leak in DRAM
- DRAM row activations cause flips



Rowhammer

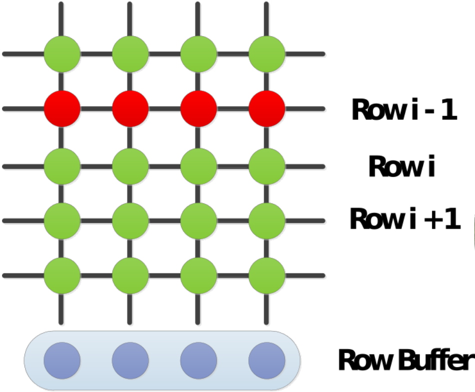
- Causes charge to leak in DRAM
- DRAM row activations cause flips





Rowhammer

- Causes charge to leak in DRAM
- DRAM row activations cause flips



Row i-1

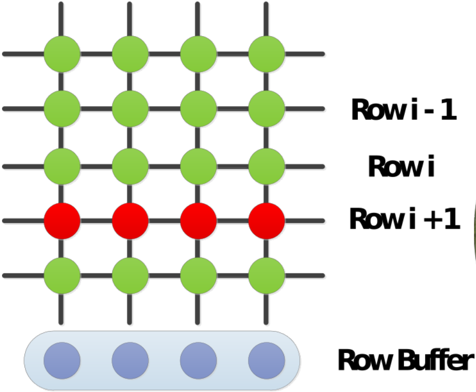
Row i

Row i+1

Row Buffer

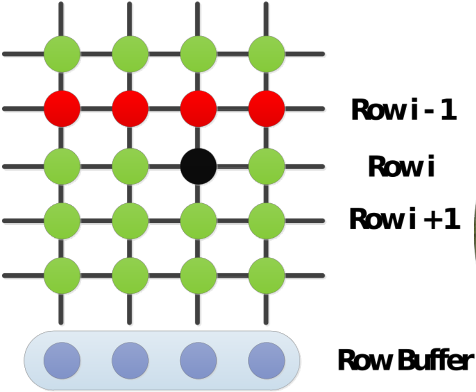
Rowhammer

- Causes charge to leak in DRAM
- DRAM row activations cause flips

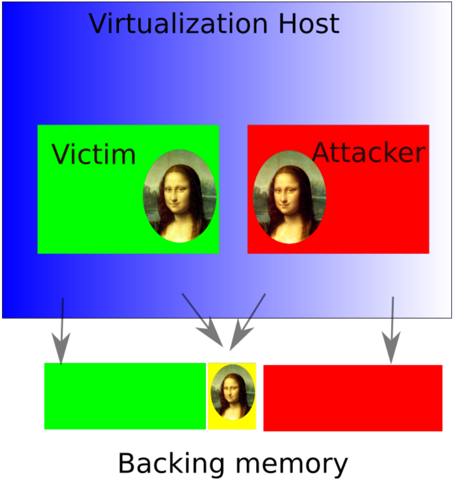


Rowhammer

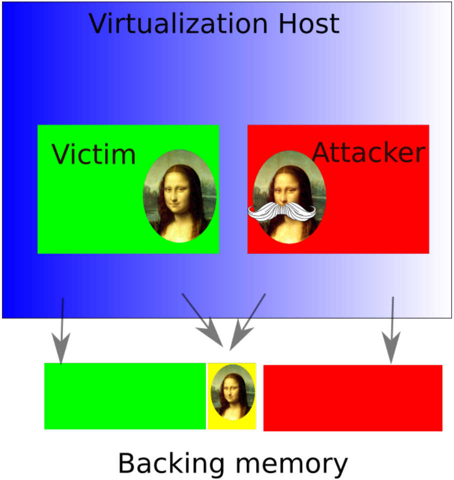
- Causes charge to leak in DRAM
- DRAM row activations cause flips



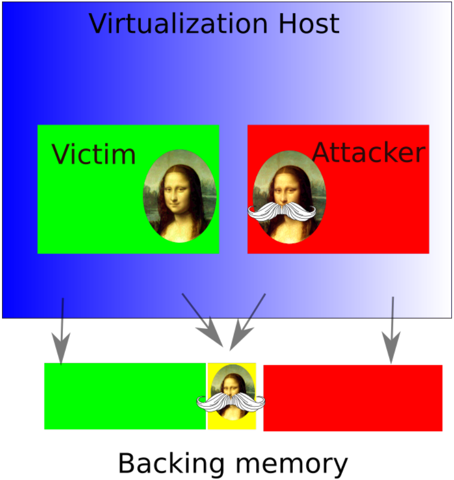
Memory deduplication + Rowhammer = FFS



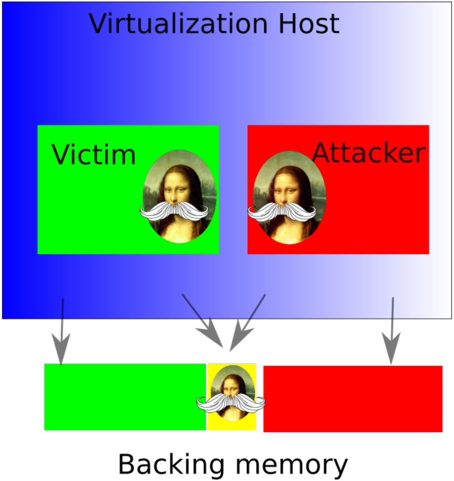
Memory deduplication + Rowhammer = FFS



Memory deduplication + Rowhammer = FFS



Memory deduplication + Rowhammer = FFS



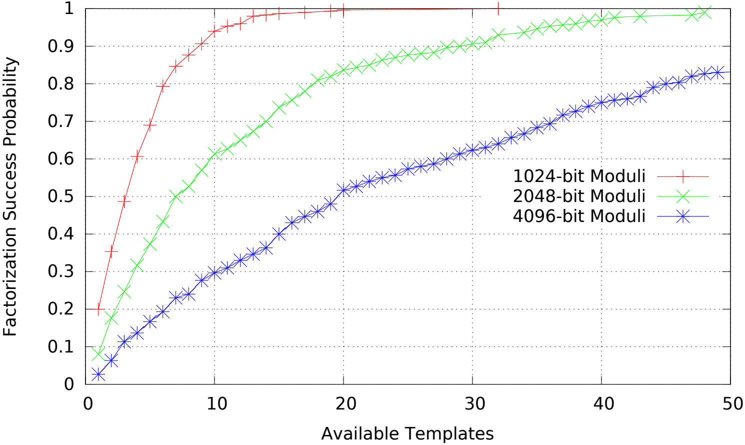
- FFS breaks COW

RSA

- Public key cryptosystem
- Two keys: public and private
- Compute secret private from factorization

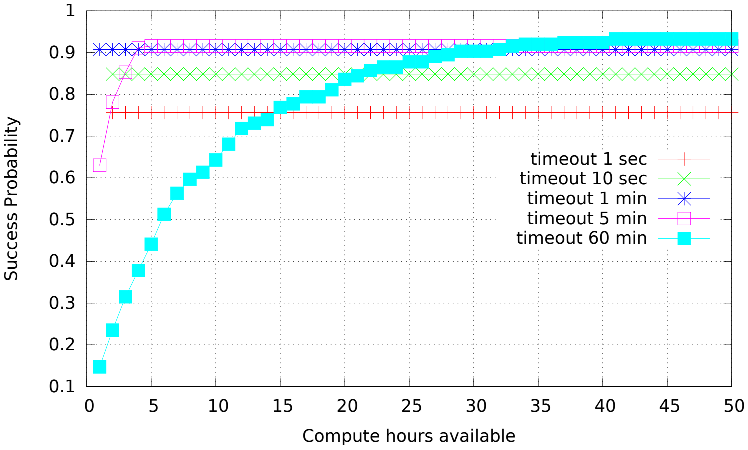
FFS - What now?

Break weakened RSA.



FFS - What now?

We can afford a short time cutoff.



Section 3

OpenSSH Attack

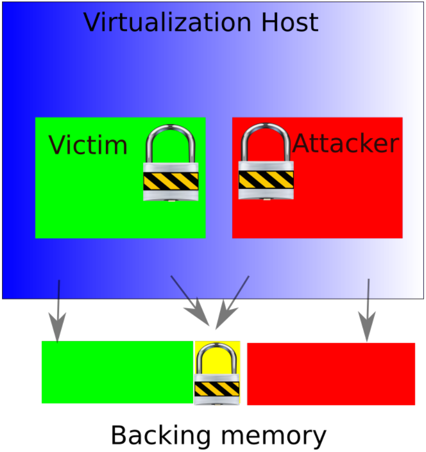
authorized_keys file

Looks like this:

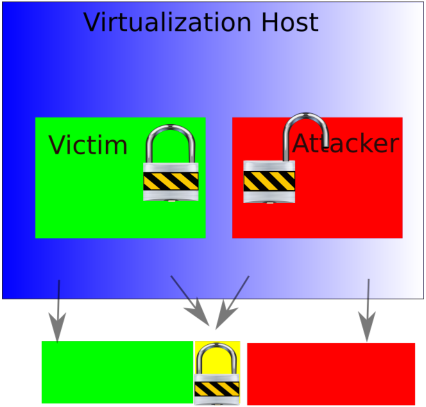
```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDX
y7MdVToVAvKB0/Xven/kqBzfRZm+GITl6sB0u+Aa
3/UTC3x+eKjB2jf+48kTP7AvsdbSwg9Q5upN77xX
3mNGwwj1RUQp0PPc99XH09M84iCydE+9smYseySf
bJQnrov5Ricz2Z18Neuy5ZUH/Ldrf1NSwWoo5NZL
6tj0E9JvZurMPPk2EqEyHltEFC60etJwEfaPq9k0
glmzFtBWLHR4dF1796JeVkFiWcmMaykAoN+JRF2n
MlayPlUxdWR0JwxZ2cJ9la/QLXvv8x0ts0RGP9ZG
5BWq0cD781evuSS3i91BNg60sl7mlxo6Mc3oUbew
/7ddV08WjdRBn7iQF9WN beng@mymachine
```

- RSA public key
- Attacker writes this to memory
- We need the private key

OpenSSH FFS attack

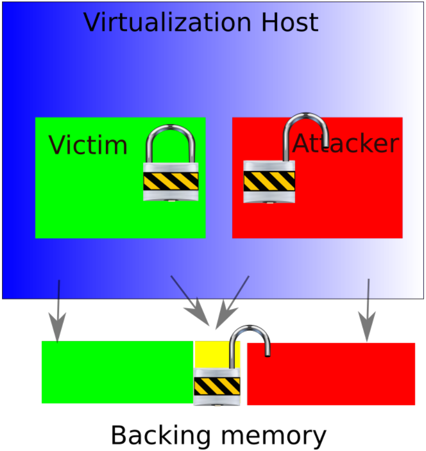


OpenSSH FFS attack

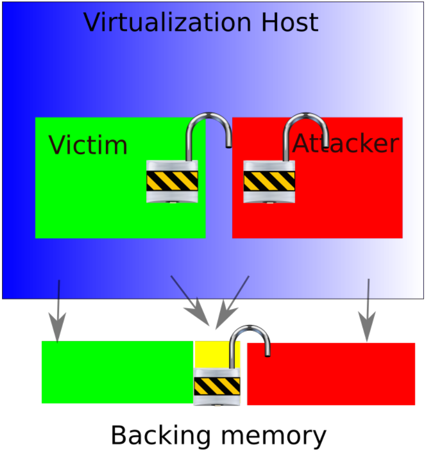


Backing memory

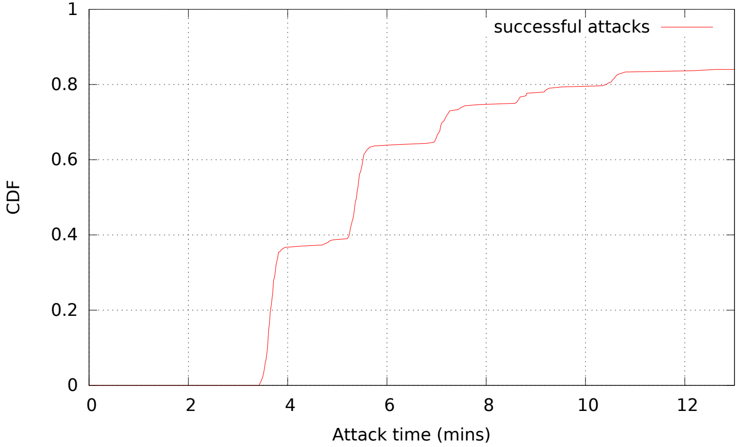
OpenSSH FFS attack



OpenSSH FFS attack



OpenSSH Attack



- Could retry

Section 4

Privilege Escalation Bitflips

What else could we bitflip

- Victim VM kernel pagetable
- On-disk victim VM inode
- Machine code

Victim VM kernel pagetable

- Linux kernel pagetables are predictable: early boot
- Mimic a kernel pagetable
- And flip the S bit
- Then we can easily upgrade our local access

On-disk victim VM inode

- Base system binaries have low variation in inode content
- Mimic a page containing an inode
- Of a small binary owned by root
- And flip the suid bit
- Then we can also easily upgrade our local access

Bitflip machine code

Original C code:

```
int verify(char *pw)
{
    if(strcmp(pw, "Secret")) return 0;
    return 1;
}

int main(int argc, char *argv[])
{
    if(verify(argv[1])) { printf("OK!\n"); }
    else { printf("Fail!\n"); return 1; }
    return 0;
}
```

Original Behaviour

```
$ ./hello asdf
Fail!
$ ./hello Secret
OK!
```

Original Assembly

```
0x02f (01) 55         PUSH RBP
0x030 (03) 4889e5     MOV RBP, RSP
0x033 (04) 4883ec10   SUB RSP, 0x10
0x037 (04) 48897df8   MOV [RBP-0x8], RDI
0x03b (04) 488b45f8   MOV RAX, [RBP-0x8]
0x03f (05) bea4064000 MOV ESI, 0x4006a4
0x044 (03) 4889c7     MOV RDI, RAX
0x047 (05) e8cdfeffff CALL 0xfffffffffffff19
0x04c (02) 85c0       TEST EAX, EAX
0x04e (02) 7407       JZ 0x57
0x050 (05) b800000000 MOV EAX, 0x0
0x055 (02) eb05       JMP 0x5c
0x057 (05) b801000000 MOV EAX, 0x1
0x05c (01) c9         LEAVE
0x05d (01) c3         RET
```

Mutated Assembly

```
0x02f (01) 55         PUSH RBP
0x030 (03) 4889e5     MOV RBP, RSP
0x033 (04) 4883e410   AND RSP, 0x10
0x037 (04) 48897df8   MOV [RBP-0x8], RDI
0x03b (04) 488b45f8   MOV RAX, [RBP-0x8]
0x03f (05) bea4064000 MOV ESI, 0x4006a4
0x044 (03) 4889c7     MOV RDI, RAX
0x047 (05) e8cdfeffff CALL 0xffffffffffffff19
0x04c (02) 85c0       TEST EAX, EAX
0x04e (02) 7407       JZ 0x57
0x050 (05) b800000000 MOV EAX, 0x0
0x055 (02) eb05       JMP 0x5c
0x057 (05) b801000000 MOV EAX, 0x1
0x05c (01) c9         LEAVE
0x05d (01) c3         RET
```

Mutated Assembly

```
0x02f (01) 55         PUSH RBP
0x030 (03) 4889e5     MOV RBP, RSP
0x033 (04) 4883e810   SUB RAX, 0x10
0x037 (04) 48897df8   MOV [RBP-0x8], RDI
0x03b (04) 488b45f8   MOV RAX, [RBP-0x8]
0x03f (05) bea4064000 MOV ESI, 0x4006a4
0x044 (03) 4889c7     MOV RDI, RAX
0x047 (05) e8cdfeffff CALL 0xffffffffffffff19
0x04c (02) 85c0       TEST EAX, EAX
0x04e (02) 7407       JZ 0x57
0x050 (05) b800000000 MOV EAX, 0x0
0x055 (02) eb05       JMP 0x5c
0x057 (05) b801000000 MOV EAX, 0x1
0x05c (01) c9         LEAVE
0x05d (01) c3         RET
```

Mutated Assembly

```
0x02f (01) 55         PUSH RBP
0x030 (03) 4889e5     MOV RBP, RSP
0x033 (04) 4883ee10   SUB RSI, 0x10
0x037 (04) 48897df8   MOV [RBP-0x8], RDI
0x03b (04) 488b45f8   MOV RAX, [RBP-0x8]
0x03f (05) bea4064000 MOV ESI, 0x4006a4
0x044 (03) 4889c7     MOV RDI, RAX
0x047 (05) e8cdfeffff CALL 0xffffffffffffff19
0x04c (02) 85c0       TEST EAX, EAX
0x04e (02) 7407       JZ 0x57
0x050 (05) b800000000 MOV EAX, 0x0
0x055 (02) eb05       JMP 0x5c
0x057 (05) b801000000 MOV EAX, 0x1
0x05c (01) c9         LEAVE
0x05d (01) c3         RET
```

Mutated Assembly

```
0x02f (01) 55         PUSH RBP
0x030 (03) 4889e5     MOV RBP, RSP
0x033 (04) 4883ed10   SUB RBP, 0x10
0x037 (04) 48897df8   MOV [RBP-0x8], RDI
0x03b (04) 488b45f8   MOV RAX, [RBP-0x8]
0x03f (05) bea4064000 MOV ESI, 0x4006a4
0x044 (03) 4889c7     MOV RDI, RAX
0x047 (05) e8cdfeffff CALL 0xffffffffffffff19
0x04c (02) 85c0       TEST EAX, EAX
0x04e (02) 7407       JZ 0x57
0x050 (05) b800000000 MOV EAX, 0x0
0x055 (02) eb05       JMP 0x5c
0x057 (05) b801000000 MOV EAX, 0x1
0x05c (01) c9         LEAVE
0x05d (01) c3         RET
```

Mutated Assembly

```
0x02f (01) 55         PUSH RBP
0x030 (03) 4889e5     MOV RBP, RSP
0x033 (04) 4883ec90   SUB RSP, -0x70
0x037 (04) 48897df8   MOV [RBP-0x8], RDI
0x03b (04) 488b45f8   MOV RAX, [RBP-0x8]
0x03f (05) bea4064000 MOV ESI, 0x4006a4
0x044 (03) 4889c7     MOV RDI, RAX
0x047 (05) e8cdfeffff CALL 0xffffffffffffff19
0x04c (02) 85c0       TEST EAX, EAX
0x04e (02) 7407       JZ 0x57
0x050 (05) b800000000 MOV EAX, 0x0
0x055 (02) eb05       JMP 0x5c
0x057 (05) b801000000 MOV EAX, 0x1
0x05c (01) c9         LEAVE
0x05d (01) c3         RET
```

Interesting case

```
0x02f (01) 55         PUSH RBP
0x030 (03) 4889e5     MOV RBP, RSP
0x033 (04) 4883ec10   SUB RSP, 0x10
0x037 (04) 48897df8   MOV [RBP-0x8], RDI
0x03b (04) 488b45f8   MOV RAX, [RBP-0x8]
0x03f (05) bea4064000 MOV ESI, 0x4006a4
0x044 (03) 4889c7     MOV RDI, RAX
0x047 (05) e8cdfeffff CALL 0xffffffffffffff19
0x04c (02) 85c0       TEST EAX, EAX
0x04e (02) 7507       JNZ 0x57
0x050 (05) b800000000 MOV EAX, 0x0
0x055 (02) eb05       JMP 0x5c
0x057 (05) b801000000 MOV EAX, 0x1
0x05c (01) c9         LEAVE
0x05d (01) c3         RET
```

New behaviour

```
$ ./out/out11567.bin Secret
Fail!
$ ./out/out11567.bin asdf
OK!
```

Section 5

GPG/APT Updates Attack Demo

GPG/APT Updates

- With FFS we flip `/etc/apt/sources.list`
- With FFS we flip `/etc/apt/trusted.gpg`
- Use computed private key
- Long term RSA Ubuntu signing keys

Section 6

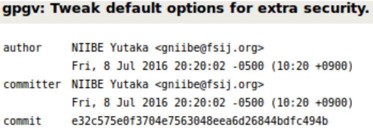
Notification, Conclusion & Further Resources

Notification

- Notified: Red Hat, Oracle, Xen, VMware, Debian, Ubuntu, OpenSSH, GnuPG, some hosting companies
- Thank you NCSC

Nationaal Cyber Security Centrum
Ministerie van Veiligheid en Justitie
Virtuele servers kunnen hun buren aanvallen

- GnuPG commit



gpgv: Tweak default options for extra security.

```
author     NIIBE Yutaka <gniibe@fsij.org>
           Fri, 8 Jul 2016 20:20:02 -0500 (10:20 +0900)
committer  NIIBE Yutaka <gniibe@fsij.org>
           Fri, 8 Jul 2016 20:20:02 -0500 (10:20 +0900)
commit     e32c575e0f3704e7563048eea6d26844bdfc494b
```

Conclusion

- Flip Feng Shui breaks isolation
- Co-hosting VMs is risky
- Disable memory dedup
- Project page

  `https://www.vusec.net/projects/flip-feng-shui`
- Want to join - PhD, postdoc, bachelor, master?

  `https://www.vusec.net/join/`