



```
addiu $sp, -0x18
sw $ra, 0x18+var_4($sp)
sw $a0, 0x18+arg_0($sp)
lui $1, 3
jal sub_2DAB8
lw $a0, dword_35A6C
lui $1, 3
lw $t7, dword_35A6C
lw $t6, dword_35A70
subu $t8, $t6, $t7
addiu $t9, $t8, 4
sltu $1, $v0, $t9
beqz $1, loc_2DA24
nop
sub_2DA28
```

Digitale Vernetzung und Verletzbarkeit von Industrie und kritischen Infrastrukturen

Bedrohungen - Schwachstellen - Abwehrstrategien

```
move $a0, $t7
lw $a0, dword_35A6C
jal sub_2DAD4
addiu $a1, $v0, 0x10
beqz $v0, loc_2DA44
move $v0, $0
la $1, dword_35A70
lw $t1, dword_35A6C
lw $t0, 0($1)
subu $t2, $t0, $t1
sra $t3, $t2, 2
sll $t4, $t3, 2
addu $t5, $v0, $t4
sw $t5, 0($1)
sw $v0, dword_35A6C
```

Invent & Verify

Intro

```

addiu $sp, -0x18
sw    $ra, 0x18+var_4($sp)
sw    $a0, 0x18+arg_0($sp)
lui   $1, 3
jal   sub_2DAB8
lw    $a0, dword_35A6C
lui   $1, 3
lw    $t7, dword_35A6C
lw    $t6, dword_35A70
subu  $t8, $t6, $t7
addiu $t9, $t7, 4
sltu  $1, $v0, $t9
beqz  $1, loc_2DA24
nop
sub  2DAAB

```

- Gregor Kopf

- Seit 2008 Security Consultant bei der Recurity Labs GmbH
- Reviews komplexer IT-Systeme und Umgebungen
 - Source Code / Binary
 - Kryptographische Protokolle

- Design

```

move  $a0, $t7
lw    $a0, dword_35A6C
jal   sub_2DAD4
addiu $a1, $v0, 0x10
beqz  $v0, loc_2DA44
move  $v0, $0
la    $1, dword_35A70
lw    $t1, dword_35A6C
lw    $t0, 0($1)
subu  $t2, $t0, $t1
sra   $t3, $t2, 2
sll   $t4, $t3, 2
addu  $t5, $v0, $t4
sw    $t5, 0($1)
sw    $v0, dword_35A6C

```

Invent & Verify



Bekannte Hacks

- TLS
 - Heartbleed
 - Diginotar
- BMW hack
- Diverse SCADA hacks
- ICANN
- New York Times (Syrian Electronic Army)

```

addiu $sp, -0x18
sw $ra, 0x18+var_4($sp)
sw $a0, 0x18+arg_0($sp)
lui $l, 3
jal sub_2DAB8
lw $a0, dword_35A6C
lui $l, 3
lw $t7, dword_35A6C
lw $t6, dword_35A70
subu $t8, $t6, $t7
addiu $t9, $t7, 4
sltu $l, $v0, $t9
beqz $l, loc_2DA24
nop
sub_2DAB8

```

```

move $a0, $t7
lw $a1, dword_35A70
jal sub_2DAB8
addiu $a1, $v0, 0x10
beqz $l, loc_2DA44
move $v0, $0
la $l, dword_35A70
lw $t1, dword_35A6C
lw $t0, 0($l)
subu $t2, $t0, $t1
sra $t3, $t2, 2
sll $t4, $t3, 2
addu $t5, $v0, $t4
sw $t5, 0($l)
sw $v0, dword_35A6C

```

Invent & Verify



Einen Schritt Zurück

- Gemeinsamkeiten
 - Vernetzung
 - Erreichbarkeit
 - Verteilung von Verantwortlichkeiten
 - Abstraktion
 - Automatisierung
 - Treffen von Annahmen

```

addiu $sp, -0x18
sw $ra, 0x18+var_4($sp)
sw $a0, 0x18+arg_0($sp)
lui $1, 3
jal sub_2DAB8
lw $a0, dword_35A6C
lui $1, 3
lw $t7, dword_35A6C
lw $t6, dword_35A70
subu $t8, $t6, $t7
addiu $t9, $t7, 4
sltu $1, $v0, $t9
beqz $1, loc_2DA24
nop
ret sub_2DA28
    
```

```

move $a0, $t7
lw $a0, dword_35A6C
jal sub_2DAD4
addiu $a1, $v0, 0x10
beqz $v0, loc_2DA44
move $v0, $0
la $1, dword_35A70
lw $t1, dword_35A6C
lw $t0, 0($1)
subu $t2, $t0, $t1
sra $t3, $t2, 2
sll $t4, $t3, 2
addu $t5, $v0, $t4
sw $t5, 0($1)
sw $v0, dword_35A6C
    
```



Vernetzung

```

addiu $sp, -0x18
sw $ra, 0x18+var_4($sp)
sw $a0, 0x18+arg_0($sp)
lui $1, 3
jal sub_2DAB8
lw $a0, dword_35A6C
lui $1, 3
lw $t7, dword_35A6C
lw $t6, dword_35A70
subu $t8, $t6, $t7
addiu $t9, $t7, 4
sllv $1, $v0, $t9
li $1, loc_2DA24
sub_2DAB8

```



```

move $a0, $t7
lw $a0, dword_35A6C
jal sub_2DAD4
addiu $a1, $v0, 0x10
beqz $v0, loc_2DA44
move $v0, $0
la $1, dword_35A70
lw $t1, dword_35A6C
lw $t0, 0($1)
subu $t2, $t0, $t1
sra $t3, $t2, 2
sll $t4, $t3, 2
addu $t5, $v0, $t4
sw $t5, 0($1)
sw $v0, dword_35A6C

```

Invent & Verify



Vernetzung

- Vormalig isolierte Systeme werden zu einem großen Netzwerk
- Alle sprechen IP
 - Autos
 - Telefone
 - Medizinische Produkte
- Nicht alle erreichbaren Parteien sind auch vertrauenswürdig
- Man kann andere erreichen, ist aber auch selbst erreichbar

```
addiu $sp, -0x18
sw $ra, 0x18+var_4($sp)
sw $a0, 0x18+arg_0($sp)
lui $1, 3
jal sub_2DAB8
lw $a0, dword_35A6C
lui $1, 3
lw $t7, dword_35A6C
lw $t6, dword_35A70
subu $t8, $t6, $t7
addiu $t9, $t7, 4
sltu $1, $v0, $t9
sub $t8, $t8, $t9
```

```
move $a0, $t7
lw $a1, sub_2DAB04
jal sub_2DAB04
addiu $a1, $v0, 0x10
beqz $v0, loc_2DA44
move $v0, $0
la $1, dword_35A70
lw $t1, dword_35A6C
lw $t0, 0($1)
subu $t2, $t0, $t1
sra $t3, $t2, 2
sll $t4, $t3, 2
addu $t5, $v0, $t4
sw $t5, 0($1)
sw $v0, dword_35A6C
```



Vernetzung

- Internet of Things
 - Und keiner weiß, welche Dinge
- M2M
- Dynamic DNS
- Autoregistration
- Mesh-Netzwerke

- Forderung nach dynamischen Systemen, die immer schwieriger zu verstehen werden

```

addiu $sp, -0x18
sw $ra, 0x18+var_4($sp)
sw $a0, 0x18+arg_0($sp)
lui $1, 3
jal sub_2DAB8
lw $a0, dword_35A6C
lui $1, 3
lw $t7, dword_35A6C
lw $t6, dword_35A70
subu $t8, $t6, $t7
addiu $t9, $t7, 4
sltu $1, $v0, $t9
beqz $1, loc_2DA24
nop
ret sub_2DAB8
    
```

```

move $a0, $t7
lw $a1, dword_35A6C
jal sub_2DAB8
addiu $a1, $v0, 0x10
beqz $v0, loc_2DA24
move $v0, $t0
la $1, dword_35A70
lw $t1, dword_35A6C
lw $t0, 0($1)
subu $t2, $t0, $t1
sra $t3, $t2, 2
sll $t4, $t3, 2
addu $t5, $v0, $t4
sw $t5, 0($1)
sw $v0, dword_35A6C
    
```

Invent & Verify



Abstraktion

```
addiu $sp, -0x18  
sw $ra, 0x18+var_4($sp)  
sw $a0, 0x18+arg_0($sp)  
lui $1, 3  
jal sub_2DAB8  
lw $a0, dword_35A6C  
lui $1, 3  
lw $t7, dword_35A6C  
lw $t6, dword_35A70  
subu $t8, $t6, $t7  
addiu $t9, $t7, 4  
stwu $1, $v0, $t9  
beoz $1, loc_2DA24
```

sub 2DAB8



```
move $a0, $t7  
lw $a0, dword_35A6C  
jal sub_2DAD4  
addiu $a1, $v0, 0x10  
beqzl $v0, loc_2DA44  
move $v0, $0  
la $1, dword_35A70  
lw $t1, dword_35A6C  
lw $t0, 0($1)  
subu $t2, $t0, $t1  
sra $t3, $t2, 2  
sll $t4, $t3, 2  
addu $t5, $v0, $t4  
sw $t5, 0($1)  
sw $v0, dword_35A6C
```

Invent & Verify



Abstraktion

```

addiu $sp, -0x18
sw    $ra, 0x18+var_4($sp)
sw    $a0, 0x18+arg_0($sp)
lui   $1, 3
jal   sub_2DAE8
lw    $a0, dword_35A6C
lui   $1, 3
lw    $t7, dword_35A6C
lw    $t6, dword_35A70
subu  $t8, $t6, $t7
addiu $t9, $t7, 4
sltu  $1, $v0, $t9
beqz  $1, loc_2DA24
nop
sub_2DAE8
    
```

- Abstraktion
 - Wiederverwenden bestehender Komponenten
 - Welche Eigenschaften dürfen angenommen werden?
 - Welche (Software-)Komponente leistet was?
- Delegieren von Verantwortung
- Leaky Abstractions
 - Ungerechtfertigte Annahmen

■ Bei Produktlebenszyklen von > 20 Jahren

```

move  $a0, $t7
lw    $a0, dword_35A6C
jal   sub_2DAD0
addiu $a1, $v0
beqz  $v0, loc_2DA44
move  $v0, $0
la    $1, dword_35A70
lw    $t1, dword_35A6C
lw    $t0, 0($1)
subu  $t2, $t0, $t1
sra   $t3, $t2, 2
sll   $t4, $t3, 2
addu  $t5, $v0, $t4
sw    $t5, 0($1)
sw    $v0, dword_35A6C
    
```



Infrastruktur

- Worauf wir aufbauen
 - BGP
 - DNS
 - Email
 - X.509
 - Authentifizierung über Google / Facebook
 - „Cloud“

▪ a.k.a. Copy Lots Of User Data

```

move $a0, $t7
lw $a0, dword_35A6C
jal sub_2DAD4
addiu $a1, $v0, 0x10
beqz $v0, loc_2DA44
move $v0, $0
la $t1, dword_35A70
lw $t1, dword_35A6C
lw $t0, 0($t1)
subu $t2, $t0, $t1
sra $t3, $t2, 2
sll $t4, $t3, 2
addu $t5, $v0, $t4
sw $t5, 0($t1)
sw $v0, dword_35A6C

```

```

addiu $sp, -0x18
sw $ra, 0x18+var_4($sp)
sw $a0, 0x18+arg_0($sp)
lui $t1, 3
jal sub_2DAB8
lw $a0, dword_35A6C
lui $t1, 3
lw $t7, dword_35A6C
lw $t6, dword_35A70
subu $t8, $t6, $t7
addiu $t9, $t7, 4
sllv $t1, $v0, $t9
beqz $t1, loc_2DA24
nop

```



Kritische Infrastruktur

```

addiu $sp, -0x18
sw $ra, 0x18+var_4($sp)
sw $a0, 0x18+arg_0($sp)
lui $1, 3
jal sub_2DAB8
lw $a0, dword_35A6C
lui $1, 3
lw $t7, dword_35A6C
lw $t6, dword_35A70
subu $t8, $t6, $t7
addiu $t9, $t7, 4
sltu $1, $v0, $t9
beqz $1, loc_2DA24
nop
sub_2DA28
    
```

- Kritische Infrastruktur

- Was ist das?

- Kritische Infrastruktur vs. kritische Anwendungen

- Wer bestimmt, was kritisch ist?

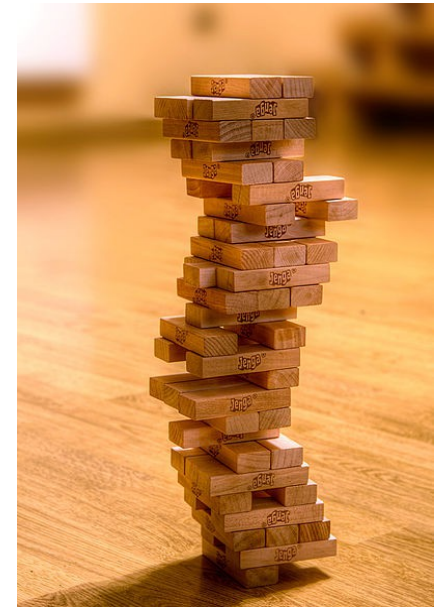
- Die Regierung (IT-Sicherheitsgesetz)?

- Frage nach Verantwortlichkeiten

- Infrastruktur kann kritisch werden

- Wenn kritische Komponenten darauf aufbauen

- Vor allem, wenn sie sich weit unten im Stack befindet



```

move $a0, $t7
lw $a0, dword_35A6C
jal sub_2DAD4
addiu $a1, $v0, 0x10
beqz $v0, loc_2DA44
move $v0, $0
la $1, dword_35A70
lw $t1, dword_35A6C
lw $t0, 0($1)
subu $t2, $t0, $t1
sra $t3, $t2, 2
sll $t4, $t3, 2
addu $t5, $v0, $t4
sw $t5, 0($1)
sw $v0, dword_35A6C
    
```

Invent & Verify



Kritische Infrastruktur

- Passwort-Reset über Email
 - MX-Records im DNS
 - Seit wann gilt Email als vertrauenswürdig?
- X.509
 - StartSSL stellt X.509-Zertifikate aus
 - Prüfung des Domaininhabers per Email
 - Mail an postmaster@, hostmaster@ oder webmaster@
 - Was könnte da wohl schiefgehen...

```

addiu $sp, -0x18
sw $ra, 0x18+var_4($sp)
sw $a0, 0x18+arg_0($sp)
lui $1, 3
jal sub_2DAB8
lw $a0, dword_35A6C
lui $1, 3
lw $t7, dword_35A6C
lw $t6, dword_35A70
subu $t8, $t6, $t7
addiu $t9, $t7, 4
sltu $1, $v0, $t9
beqz $1, loc_2DA24
nop
sub_2DA28

```

```

move $a0, $t7
lw $a0, dword_35A6C
jal sub_2DAD4
addiu $a1, $v0, 0x10
beqz $v0, loc_2DA44
move $v0, $0
la $1, dword_35A70
lw $t1, dword_35A6C
lw $t0, 0($1)
subu $t2, $t0, $t1
sra $t3, $t2, 2
sll $t4, $t3, 2
addu $t5, $v0, $t4
sw $t5, 0($1)
sw $v0, dword_35A6C

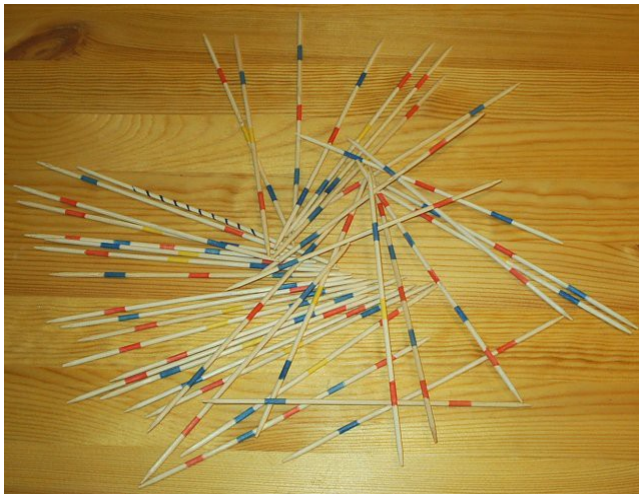
```



gTLD

- Grundlegende Änderung in bestehender Komponente
 - Meine Wunschliste: .onion, .daimler, .scada, .bmw
- GccTLD
 - Google interpretiert einige ccTLD mit anderer Semantik

```
addiu $sp, -0x18
sw $ra, 0x18+var_4($sp)
sw $a0, 0x18+arg_0($sp)
lui $1, 3
jal sub_2DAB8
lw $a0, dword_35A6C
lui $1, 3
lw $t7, dword_35A6C
lw $t6, dword_35A70
subu $t8, $t6, $t7
addiu $t9, $t7, 4
sltu $1, $v0, $t9
beqz $1, loc_2DA24
nop
ret sub_2DAB8
```



```
move $t3, $t2, 2
sll $t4, $t3, 2
addu $t5, $v0, $t4
sw $t5, 0($1)
sw $v0, dword_35A6C
```

Invent & Verify



gTLD

- Ungerechtfertigte Annahme: .onion ist keine gültige TLD
 - Welches RFC schreibt die Abwesenheit von .onion vor?
 - Analogie: RFC1918-Adressen
 - 2.2.2.2 als interne Adresse verwenden?
 - Hallo, DTAG
- Besserer Ansatz
 - ISO 3166-1: User-assigned code elements
- Das Problem resultiert aus ungültigen Annahmen

```
move $a0, $t7
lw $a0, dword_35A6C
jal sw $a0, 0x18+var_4($sp)
addiu $a1, $v0, 0x18
beqz $v0, loc_2DA44
move $v0, $0
la $t1, dword_35A70
lw $t1, dword_35A6C
lw $t0, 0($t1)
subu $t2, $t0, $t1
sra $t3, $t2, 2
sll $t4, $t3, 2
addu $t5, $v0, $t4
sw $t5, 0($t1)
sw $v0, dword_35A6C
```

```
addiu $sp, -0x18
sw $ra, 0x18+var_4($sp)
sw $a0, 0x18+arg_0($sp)
lui $t1, 3
jal sub_2DAB8
lw $a0, dword_35A6C
lui $t1, 3
lw $t7, dword_35A6C
lw $t6, dword_35A70
subu $t8, $t6, $t7
addiu $t9, $t7, 4
sllv $t1, $v0, $t9
beqz $t1, loc_2DA24
nop
sub 2DA48
```



Probleme

```

addiu $sp, -0x18
sw    $ra, 0x18+var_4($sp)
sw    $a0, 0x18+arg_0($sp)
lui   $1, 3
jal   sub_2DAB8
lw    $a0, dword_35A6C
lui   $1, 3
lw    $t7, dword_35A6C
lw    $t6, dword_35A70
subu  $t8, $t6, $t7
addiu $t9, $t7, 4
sltu  $1, $v0, $t9
beqz  $1, loc_2DA24
nop
sub_2DA28
    
```

- Fehlerhafte Annahmen
 - Der Angreifer kontrolliert das DNS nicht
 - Im Internet mag das (manchmal) stimmen
 - Am Flughafen / im Hotel auch?
 - .gibt-es-nicht ist keine gültige TLD
 - Und 2.2.2.2 kann man problemlos intern verwenden...
- Auf der positiven Seite:

OpenSSH kann Host-Keys per DNS prüfen

Fragt standardmäßig dennoch nach (außer bei DNSSEC)

```

move  $a0, $t7
lw    $a0, dword_35A6C
jal   sub_2DAD0
addiu $a1, $v0, 0x10
beqz  $v0, loc_2DA44
move  $v0, $t0
la    $1, dword_35A6C
lw    $t1, dword_35A6C
lw    $t0, 0($1)
subu  $t2, $t0, $t1
sra   $t3, $t2, 2
sll   $t4, $t3, 2
addu  $t5, $v0, $t4
sw    $t5, 0($1)
sw    $v0, dword_35A6C
    
```

Invent & Verify



Und die Sicherheit?

```

addiu $sp, -0x18
sw $ra, 0x18+var_4($sp)
sw $a0, 0x18+arg_0($sp)
lui $l, 3
jal sub_2DAB8
lw $a0, dword_35A6C
lui $l, 3
lw $t7, dword_35A6C
lw $t6, dword_35A70
subu $t8, $t6, $t7
addiu $t9, $t7, 4
sltu $l, $v0, $t9

```

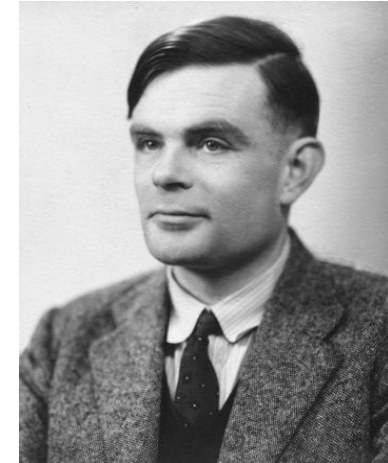
- Wie funktioniert Security?

- Die bauen wir oben drauf!

- AV
 - IDS
 - IPS

- Grundlegende Probleme ignorieren wir dabei

- Und unseren schiefen Turm stützen wir seitlich ab



```

move $a0, $t7
lw $a0, dword_35A6C
jal sub_2DAD4
addiu $a1, $v0, 0x10
beqz $v0, loc_2DA44
move $v0, $0
la $l, dword_35A70
lw $t1, dword_35A6C
lw $t0, 0($l)
subu $t2, $t0, $t1
sra $t3, $t2, 2
sll $t4, $t3, 2
addu $t5, $v0, $t4
sw $t5, 0($l)
sw $v0, dword_35A6C

```

- ASLR
- DEP

Invent & Verify



Sicherheit

```

addiu $sp, -0x18
sw $ra, 0x18+var_4($sp)
sw $a0, 0x18+arg_0($sp)
lui $1, 3
jal sub_2DAB8
lw $a0, dword_35A6C
lui $1, 3
lw $t7, dword_35A6C
lw $t6, dword_35A70
subu $t8, $t6, $t7
addiu $t9, $t7, 4
stwu $1, $v0, $t9
and $1, $t9, 24
sub 2DAAB
    
```

- Sicherheit ist charakterisiert durch nicht-funktionale Eigenschaften

- Beispiel:

- Anwesenheit von Zugriffskontrolle
- Abwesenheit von
 - Speicherkorruption
 - Injection-Problemen
 - Kryptographischen Seitenkanälen

- Kein „one size fits it all“

```

move $a0, $t7
lw $a0, dword_35A6C
jal sub_2DAD4
addiu $a1, $v0, 10
beqz $v0, loc_2DA44
move $v0, $0
la $1, dword_35A70
lw $t1, dword_35A6C
lw $t0, 0($1)
subu $t1, $t1, $t0
sra $t2, $t1, 2
sll $t4, $t3, 2
addu $t5, $v0, $t4
sw $t5, 0($1)
sw $v0, dword_35A6C
    
```

Invent & Verify



Zusammenfassung

- Sicherheitseigenschaften werden behauptet oder angenommen
 - Wo ist die Produkthaftung?
- Alte Weisheit: man muss auch das Auto vor und hinter sich fahren
 - Aufbau sicherer Systeme ist nicht anders
 - Akteure sollten sowohl den Stack unterhalb als auch oberhalb des eigenen Systems kennen
- In Mitarbeiter investieren!

```
addiu $sp, -0x18
sw $ra, 0x18+var_4($sp)
sw $a0, 0x18+arg_0($sp)
lui $1, 3
jal sub_2DAB8
lw $a0, dword_35A6C
lui $1, 3
lw $t7, dword_35A6C
lw $t6, dword_35A70
subu $t8, $t6, $t7
addiu $t9, $t7, 4
sltu $1, $v0, $t9
jal sub_2DA24
sub 3DAAB
```

```
move $a0, $t7
lw $a0, dword_35A6C
jal sub_2DAD4
addiu $a1, $v0, 0x10
beqz $v0, loc_2DA44
move $v0, $0
la $1, dword_35A70
lw $t1, dword_35A6C
lw $t0, 0($1)
subu $t2, $t0, $t1
sra $t3, $t2, 2
sll $t4, $t3, 2
addu $t5, $v0, $t4
sw $t5, 0($1)
sw $v0, dword_35A6C
```

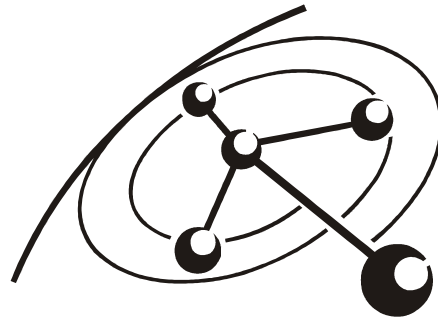


Vielen Dank.

```

addiu $sp, -0x18
sw $ra, 0x18+var_4($sp)
sw $a0, 0x18+arg_0($sp)
lui $1, 3
jal sub_2DAB8
lw $a0, dword_35A6C
lui $1, 3
lw $t7, dword_35A6C
lw $t6, dword_35A70
subu $t8, $t6, $t7
addiu $t9, $t7, 4
stwu $1, $v0, $t9
beqz $1, loc_2DA24
NOP
sub_2DAB8

```



Recurity Labs

Gregor Kopf
Security Consultant

greg@recurity-labs.com

Recurity Labs GmbH, Berlin, Germany
<http://www.recurity-labs.com>

```

move $a0, $t7
lw $a0, dword_35A6C
jal sub_2DAD4
addiu $a1, $v0, 0x10
beqz $v0, loc_2DA44
move $v0, $0
la $1, dword_35A70
lw $t1, dword_35A6C
lw $t0, 0($1)
subu $t2, $t0, $t1
sra $t3, $t2, 2
sll $t4, $t3, 2
addu $t5, $v0, $t4
sw $t5, 0($1)
sw $v0, dword_35A6C

```

Invent & Verify

