On Subverting Trust

for NDSS 2016

Matthew Green
Johns Hopkins University

Reflections on Trusting Trust

To what extent should one trust a statement that a program is free of Trojan horses? Perhaps it is more important to trust the people who wrote the software.

KEN THOMPSON

INTRODUCTION

I thank the ACM for this award. I can't help but feel that I am receiving this honor for timing and serendipity as much as technical merit. UNIX[1] swept into popularity with an industry-wide change from central mainframes to autonomous minis. I suspect that Daniel Bobrow [1] would be here instead of me if he could afford a PDP-10 and had used it more than I. Moreover, programs. I would like to present to you the cutest program I ever wrote. I will do this in three stages and try to bring it together at the end.

STAGE I

ScreenOS Version	Released	CVE-2015-7755 (telnet/ssh)	CVE-2015-7756 (VPN)
6.2.0r15		not vulnerable	vulnerable
6.2.0r16	March 2013	not vulnerable	vulnerable
6.2.0r17	May 2013	not vulnerable	vulnerable
6.2.0r18	Oct 2013	not vulnerable	vulnerable
6.3.0r12	Aug 2012	not vulnerable	vulnerable
6.3.0r13	Dec 2012	not vulnerable	vulnerable
6.3.0r14	Apr 2013	not vulnerable	vulnerable
6.3.0r15	Sep 2013	not vulnerable	vulnerable
6.3.0r16	Dec 2013	not vulnerable	vulnerable
6.3.0r17	Apr 2014	vulnerable	vulnerable
6.3.0r18	Dec 2014	vulnerable	vulnerable
6.3.0r19	May 2015	vulnerable	vulnerable
6.3.0r20		vulnerable	vulnerable
6.3.0r21	Dec 2015	not vulnerable	not vulnerable

Source: CertBund

Source: NYT/ProPublica

TOP SECRET//SI/TK//NOFORN

(U) COMPUTER NETWORK OPERATIONS
(U) SIGINT ENABLING

(U) Project Description

(TS//SI//NF) The SIGINT Enabling Project actively engages the US and foreign IT industries to covertly influence and/or overtly leverage their commercial products' designs. These design changes make the systems in question exploitable through SIGINT collection (e.g., Endpoint, MidPoint, etc.) with foreknowledge of the modification. To the consumer and other adversaries, however, the systems' security remains intact. In this

- (TS//SI//NF) Shape the worldwide commercial cryptography marketplace to make it more tractable to advanced cryptanalytic capabilities being developed by NSA/CSS. [CCP_00090]

- (TS//SI//REL TO USA, FVEY) Insert vulnerabilities into commercial encryption systems, IT systems, networks, and endpoint communications devices used by targets.
- (TS//SI//REL TO USA, FVEY) Collect target network data and metadata via cooperative network carriers and/or increased control over core networks.
- (TS//SI//REL TO USA, FVEY) Leverage commercial capabilities to remotely deliver or receive information to and from target endpoints.
- (TS//SI//REL TO USA, FVEY) Exploit foreign trusted computing platforms and technologies.
- (TS//SI//REL TO USA, FVEY) Influence policies, standards and specification for commercial public key technologies.

A CRYPTO NERD'S IMAGINATION:

HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.

NO GOOD! IT'S 4096-BIT RSA!

BLAST! OUR EVIL PLAN IS FOILED!

WHAT WOULD ACTUALLY HAPPEN:

HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.

GOT IT.

TOP SECRET STRAP1

Response to improving security

- For the past decade, NSA has lead an aggressive, multi-pronged effort to break widely used Internet encryption technologies
- Cryptanalytic capabilities are now coming on line
- Vast amounts of encrypted Internet data which have up till now been discarded are now exploitable
- Major new processing systems, SIGDEV efforts and tasking must be put in place to capitalize on this opportunity

PTD "We penetrate targets' defences."

This talk

- Examples of systems that, if not subverted, have the hallmarks of being so
  - In one case, subversion is covert
  - In another, it is overt (courts, congress)
- And a bit on what we might be able to do about it
- **Main case study: Juniper ScreenOS**

Includes work by:

Ralf Phillip Weinmann, William Pinckaers, Stephen Checkoway, Hovav Shacham, Eric Rescorla, Adam Langley, Nadia Heninger, Shanaan Cohney, H.D. Moore, Jake Masckeiwicz *and others*…

Juniper is committed to maintaining the integrity and security of our products and wanted to make customers aware of critical patched releases we are issuing today to address vulnerabilities in devices running ScreenOS® software.

During a recent internal code review, Juniper discovered unauthorized code in ScreenOS that could allow a knowledgeable attacker to gain administrative access to NetScreen® devices and to decrypt VPN connections. Once we identified these vulnerabilities, we launched an investigation into the matter, and worked to develop and issue patched releases for the latest versions of ScreenOS.

At this time, we have not received any reports of these vulnerabilities being exploited; however, we strongly recommend that customers update their systems and apply the patched releases with the highest priority.

ScreenOS Version	Released	CVE-2015-7755 (telnet/ssh)	CVE-2015-7756 (VPN)
6.2.0r15		not vulnerable	vulnerable
6.2.0r16	March 2013	not vulnerable	vulnerable
6.2.0r17	May 2013	not vulnerable	vulnerable
6.2.0r18	Oct 2013	not vulnerable	vulnerable
6.3.0r12	Aug 2012	not vulnerable	vulnerable
6.3.0r13	Dec 2012	not vulnerable	vulnerable
6.3.0r14	Apr 2013	not vulnerable	vulnerable
6.3.0r15	Sep 2013	not vulnerable	vulnerable
6.3.0r16	Dec 2013	not vulnerable	vulnerable
6.3.0r17	Apr 2014	vulnerable	vulnerable
6.3.0r18	Dec 2014	vulnerable	vulnerable
6.3.0r19	May 2015	vulnerable	vulnerable
6.3.0r20		vulnerable	vulnerable
6.3.0r21	Dec 2015	not vulnerable	not vulnerable

Source: CertBund

CVE-2015-7755

```
                STR             R12, [SP,#0x30+var_24]
                LDR             R0, =aSCtUUnSSipSDip ; ">>> %s(ct=%u, un='%s',
                LDR             R1, =aAuth_admin_int ; "auth_admin_internal"
                BL              sub_558F74

_13DC5C                                 ; CODE XREF: auth_admin_internal+2C↑j
                ADD             R0, R5, #0x44
                LDR             R1, =aSUnSU ; "<<< %s(un='%s') = %u"
                BL              strcmp
                CMP             R0, #0
                BNE             loc_13DC78
                MOV             R0, #0xFFFFFFFD
                LDMDB           R11, {R4-R8,R11,SP,PC}
-------------------------------------------------------------------------------

_13DC78                                 ; CODE XREF: auth_admin_internal+80↑j
                ADD             R0, R5, #0x6C
                BL              sub_14724C
                MOVS            R0, R0,LSL#16
```

<<< %s(un='%s') = %u

Source: Rapid7

CVE-2015-7755

<<< %s(un='%s') = %u

Test Your Password		Minimum Requirements
Password:	••••••••••••••••••••••	• Minimum 8 characters in length • Contains 3/4 of the following items: - Uppercase Letters - Lowercase Letters - Numbers - Symbols
Hide:	☑	
Score:	100%	
Complexity:	Very Strong	

Source: passwordmeter.com

ScreenOS Version	Released	CVE-2015-7755 (telnet/ssh)	CVE-2015-7756 (VPN)
6.2.0r15		not vulnerable	vulnerable
6.2.0r16	March 2013	not vulnerable	vulnerable
6.2.0r17	May 2013	not vulnerable	vulnerable
6.2.0r18	Oct 2013	not vulnerable	vulnerable
6.3.0r12	Aug 2012	not vulnerable	vulnerable
6.3.0r13	Dec 2012	not vulnerable	vulnerable
6.3.0r14	Apr 2013	not vulnerable	vulnerable
6.3.0r15	Sep 2013	not vulnerable	vulnerable
6.3.0r16	Dec 2013	not vulnerable	vulnerable
6.3.0r17	Apr 2014	vulnerable	vulnerable
6.3.0r18	Dec 2014	vulnerable	vulnerable
6.3.0r19	May 2015	vulnerable	vulnerable
6.3.0r20		vulnerable	vulnerable
6.3.0r21	Dec 2015	not vulnerable	not vulnerable

Source: CertBund

CVE-2015-**7756**

VPN Decryption (CVE-2015-7756) may allow a knowledgeable attacker who can monitor VPN traffic to decrypt that traffic. It is independent of the first issue.

This issue affects ScreenOS 6.2.0r15 through 6.2.0r18 and 6.3.0r12 through 6.3.0r20. **No other Juniper products or versions of ScreenOS are affected by this issue.**

There is no way to detect that this vulnerability was exploited.

This issue has been assigned CVE-2015-7756.

VPN Decryption (CVE-2015-7556) may allow a knowledgeable attacker who can monitor VPN traffic to decrypt that traffic.

CVE-2015-7756

Matthew Green
@matthew_d_green

Follow

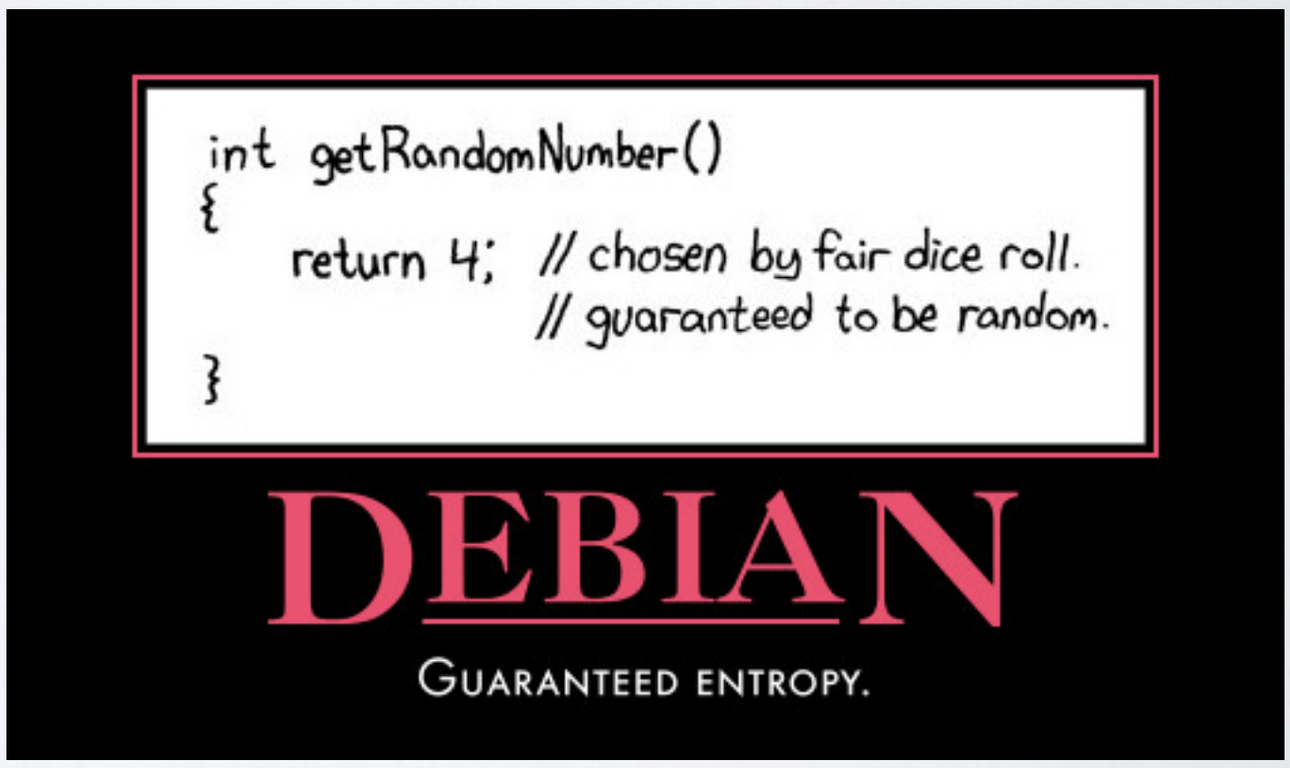
I'm really invested in the idea that this Juniper encryption vulnerability is going to be amazing. Like, Flame-level amazing.

10:30 PM - 17 Dec 2015

51 70

goto fail; // Apple SSL bug test site

This site will help you determine whether your computer is vulnerable to #goto



Vulnerable -> Patched

ScreenOS 6.3.0r20

(vulnerable)

```
2551....9585320EEAF81044F20D5503
0A035B11BECE81C785E6C933E4A8A131
F6578107....interrupt disabled a
2551....2c55e5e45edf713dc43475ef
fe8813a60326a64d9ba3d2e39cb639b0
f3b0ad10....interrupt disabled a
```

ScreenOS 6.3.0r21

(patched)

Sources: Adam Caudill, Peter Bowen, HD Moore, Ralf Phillip Weinmann

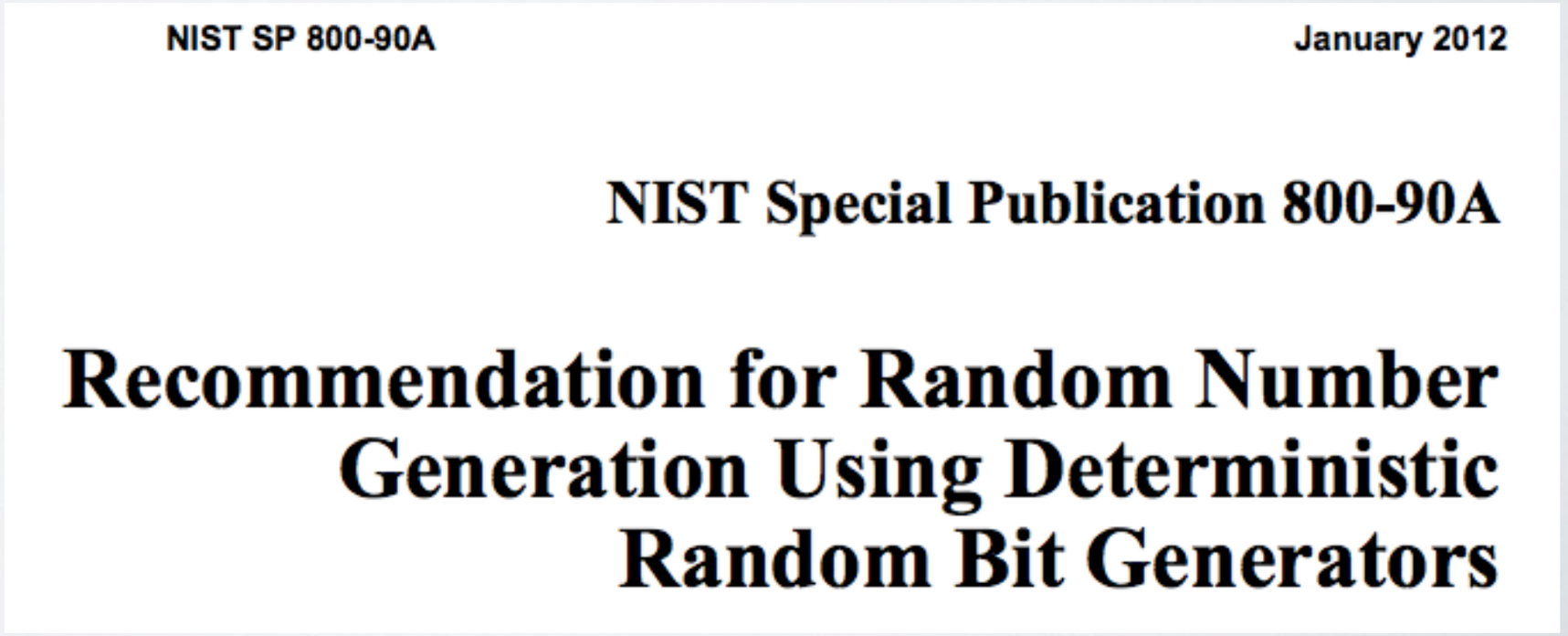
Dual EC DRBG

P-256 Weierstraß *b*

P-256 *P* x coord

P-256 field order

```
      5AC635D8AA3A93E7B3EBBD55769886BC651D06B                C3E27D2604B
      6B17D1F2E12C4247F8BCE6E563A440F277037D812DEB33A0F4A139
      FFFFFFFF00000000FFFFFFFFFFFFFFFFBCE6FAADA7179E84F3B9CAC2FC632551
bad:  9585320EEAF81044F20D55030A035B11BECE81C785E6C933E4A8A131F6578107
good: 2c55e5e45edf713dc43475effe8813a60326a64d9ba3d2e39cb639b0f3b0ad10
nist: c97445f45cdef9f0d3e05e1e585fc297235b82b5be8ff3efca67c59852018192
```



NIST SP 800-90A January 2012

NIST Special Publication 800-90A

Recommendation for Random Number Generation Using Deterministic Random Bit Generators

Dual EC DRBG

- PRNG based on EC scalar must
  - Standardized through NIST, ANSI, ISO
- **CRYPTO 2007**: Shumow, Ferguson (MSR) warn about possible backdoor
- Allows full state recovery when attacker selects point Q
- Prediction of all subsequent outputs given >27 bytes of output

N.S.A. Able to Foil Basic Safeguards of Privacy on Web

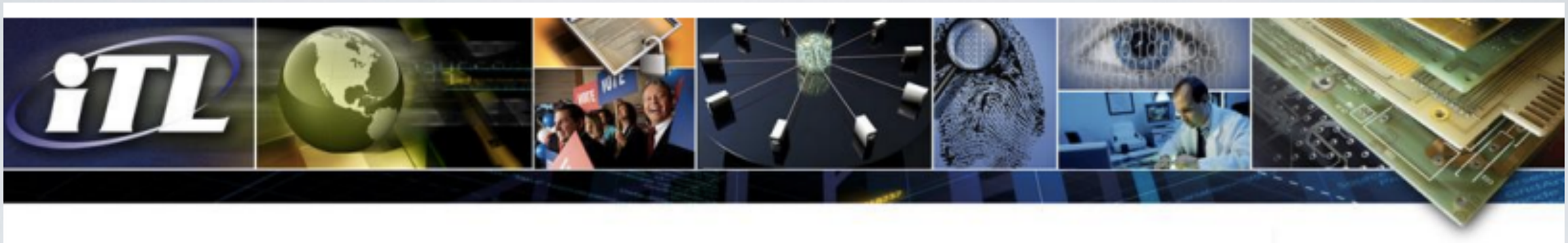
By NICOLE PERLROTH, JEFF LARSON and SCOTT SHANE
Published: September 5, 2013 | 1466 Comments

Cryptographers have long suspected that the agency planted vulnerabilities in a standard adopted in 2006 by the National Institute of Standards and Technology and later by the International Organization for Standardization, which has 163 countries as members.

Classified N.S.A. memos appear to confirm that the fatal weakness, discovered by two Microsoft cryptographers in 2007, was engineered by the agency. The N.S.A. wrote the standard and aggressively pushed it on the international group, privately calling the effort "a challenge in finesse."

"Eventually, N.S.A. became the sole editor," the memo says.

Dual EC DRBG



SUPPLEMENTAL ITL BULLETIN FOR SEPTEMBER 2013

NIST works to publish the strongest cryptographic standards possible, and uses a transparent, public process to rigorously vet its standards and guidelines. If vulnerabilities are found, NIST works with the cryptographic community to address them as quickly as possible.

In light of the concerns expressed regarding Dual_EC_DRBG, ITL is taking the following actions:

Recommending against the use of SP 800-90A Dual Elliptic Curve Deterministic Random Bit Generation: NIST strongly recommends that, pending the resolution of the security concerns and the re-issuance of SP 800-90A, the Dual_EC_DRBG, as specified in the January 2012 version of SP 800-90A, no longer be used.

FIPS Approved Algorithms

The following FIPS approved algorithms are supporte

- DSA , ECDSA Sign Verify
- SHA-1, SHA-256
- Triple-DES (CBC)

FIPS 140-2 SECURITY POLICY

Juniper Networks, Inc.

SSG 5 and SSG 20

HW P/N SSG-5 and SSG-20, FW Version ScreenOS 6.2.0

Document # 530-023728-01

Juniper Networks SSG 5 and SSG 20 Security Policy

- AES (CBC)
- HMAC-SHA-1, HMAC-SHA-256
- RSA Sign/Verify (PKCS #1)
- ANSI X9.31 DRNG

Juniper doesn't appear to use Dual EC…

Dual EC in ScreenOS

The following product families do utilize Dual_EC_DRBG, but do not use the pre-defined points cited by NIST:

1. ScreenOS*

*ScreenOS does make use of the Dual_EC_DRBG standard, but is designed to not use Dual_EC_DRBG as its primary random number generator. ScreenOS uses it in a way that should not be vulnerable to the possible issue that has been brought to light. Instead of using the NIST recommended curve points it uses self-generated basis points and then takes the output as an input to FIPS/ANSI X.9.31 PRNG, which is the random number generator used in ScreenOS cryptographic operations.

"ScreenOS does make use of the Dual_EC_DRBG standard, but is designed not to use Dual_EC_DRBG as its primary random number generator. ScreenOS uses it it in a way that shouldn't be vulnerable to the possible issue that has been brought to light." (2013)

Dual EC in ScreenOS

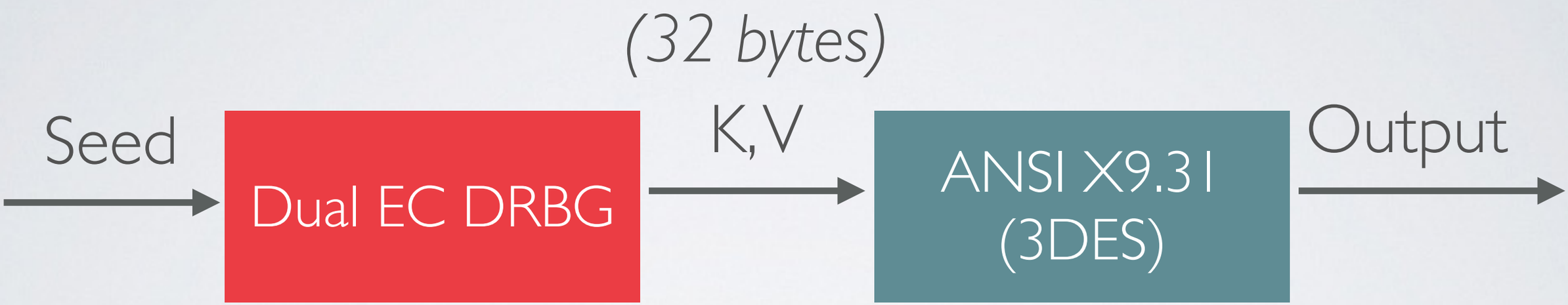
The following product families do utilize Dual_EC_DRBG, but do not use the pre-defined points cited by NIST:

1. ScreenOS*

*ScreenOS does make use of the Dual_EC_DRBG standard, but is designed to not use Dual_EC_DRBG as its primary random number generator. ScreenOS uses it in a way that should not be vulnerable to the possible issue that has been brought to light. Instead of using the NIST recommended curve points it uses self-generated basis points and then takes the output as an input to FIPS/ANSI X.9.31 PRNG, which is the random number generator used in ScreenOS cryptographic operations.

"ScreenOS does make use of the Dual_EC_DRBG standard, but is designed not to use Dual_EC_DRBG as its primary random number generator. ScreenOS uses it it in a way that shouldn't be vulnerable to the possible issue that has been brought to light." (2013)

RNG Cascade



This approach *should* neutralize any backdoor

```
1  void prng_reseed(void) {
2    blocks_generated_since_reseed = 0;
3    if (dualec_generate(prng_temporary, 32) != 32)
4      error_handler("FIPS ERROR: PRNG failure, unable to reseed\n", 11);
5    memcpy(prng_seed, prng_temporary, 8);
6    prng_output_index = 8;
7    memcpy(prng_key, &prng_temporary[prng_output_index], 24);
8    prng_output_index = 32;
9  }
10 void prng_generate(void) {
11   int time[2];
12
13   time[0] = 0;
14   time[1] = get_cycles();
15   prng_output_index = 0;
16   ++blocks_generated_since_reseed;
17   if (!one_stage_rng())
18     prng_reseed();
19   for (; prng_output_index <= 0x1F; prng_output_index += 8) {
20     // FIPS checks removed for clarity
21     x9_31_generate_block(time, prng_seed, prng_key, prng_block);
22     // FIPS checks removed for clarity
23     memcpy(&prng_temporary[prng_output_index], prng_block, 8);
24   }
25 }
```

Calls Dual EC to fill a buffer

Credit: William Pinckaers

```
1  void prng_reseed(void) {
2    blocks_generated_since_reseed = 0;
3    if (dualec_generate(prng_temporary, 32) != 32)
4      error_handler("FIPS ERROR: PRNG failure, unable to reseed\n", 11);
5    memcpy(prng_seed, prng_temporary, 8);
6    prng_output_index = 8;
7    memcpy(prng_key, &prng_temporary[prng_output_index], 24);
8    prng_output_index = 32;
9  }
10 void prng_generate(void) {
11   int time[2];
12
13   time[0] = 0;
14   time[1] = get_cycles();
15   prng_output_index = 0;
16   ++blocks_generated_since_reseed;
17   if (!one_stage_rng())
18     prng_reseed();
19   for (; prng_output_index <= 0x1F; prng_output_index += 8) {
20     // FIPS checks removed for clarity
21     x9_31_generate_block(time, prng_seed, prng_key, prng_block);
22     // FIPS checks removed for clarity
23     memcpy(&prng_temporary[prng_output_index], prng_block, 8);
24   }
25 }
```

"Runs" ANSI generator in place

Credit: William Pinckaers

```
1  void prng_reseed(void) {
2    blocks_generated_since_reseed = 0;
3    if (dualec_generate(prng_temporary, 32) != 32)
4      error_handler("FIPS ERROR: PRNG failure, unable to reseed\n", 11);
5    memcpy(prng_seed, prng_temporary, 8);
6    prng_output_index = 8;
7    memcpy(prng_key, &prng_temporary[prng_output_index], 24);
8    prng_output_index = 32;
9  }
10 void prng_generate(void) {
11   int time[2];
12
13   time[0] = 0;
14   time[1] = get_cycles();
15   prng_output_index = 0;
16   ++blocks_generated_since_reseed;
17   if (!one_stage_rng())
18     prng_reseed();
19   for (; prng_output_index <= 0x1F; prng_output_index += 8) {
20     // FIPS checks removed for clarity
21     x9_31_generate_block(time, prng_seed, prng_key, prng_block);
22     // FIPS checks removed for clarity
23     memcpy(&prng_temporary[prng_output_index], prng_block, 8);
24   }
25 }
```

Generates Dual EC output

Sets prng_output_index = 32

Credit: William Pinckaers

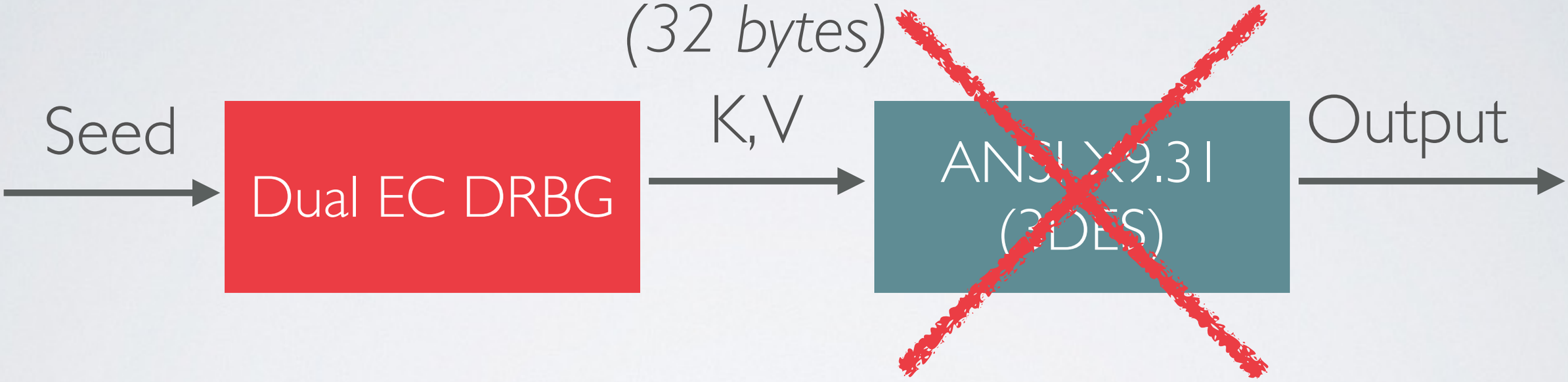
```
1  void prng_reseed(void) {
2    blocks_generated_since_reseed = 0;
3    if (dualec_generate(prng_temporary, 32) != 32)
4      error_handler("FIPS ERROR: PRNG failure, unable to reseed\n", 11);
5    memcpy(prng_seed, prng_temporary, 8);
6    prng_output_index = 8;
7    memcpy(prng_key, &prng_temporary[prng_output_index], 24);
8    prng_output_index = 32;
9  }
10 void prng_generate(void) {
11   int time[2];
12
13   time[0] = 0;
14   time[1] = get_cycles();
15   prng_output_index = 0;
16   ++blocks_generated_since_reseed;
17   if (!one_stage_rng())
18     prng_reseed();
19   for (; prng_output_index <= 0x1F; prng_output_index += 8) {
20     // FIPS checks removed for clarity
21     x9_31_generate_block(time, prng_seed, prng_key, prng_block);
22     // FIPS checks removed for clarity
23     memcpy(&prng_temporary[prng_output_index], prng_block, 8);
24   }
25 }
```

ANSI generator is never run.

Dual EC output emitted.

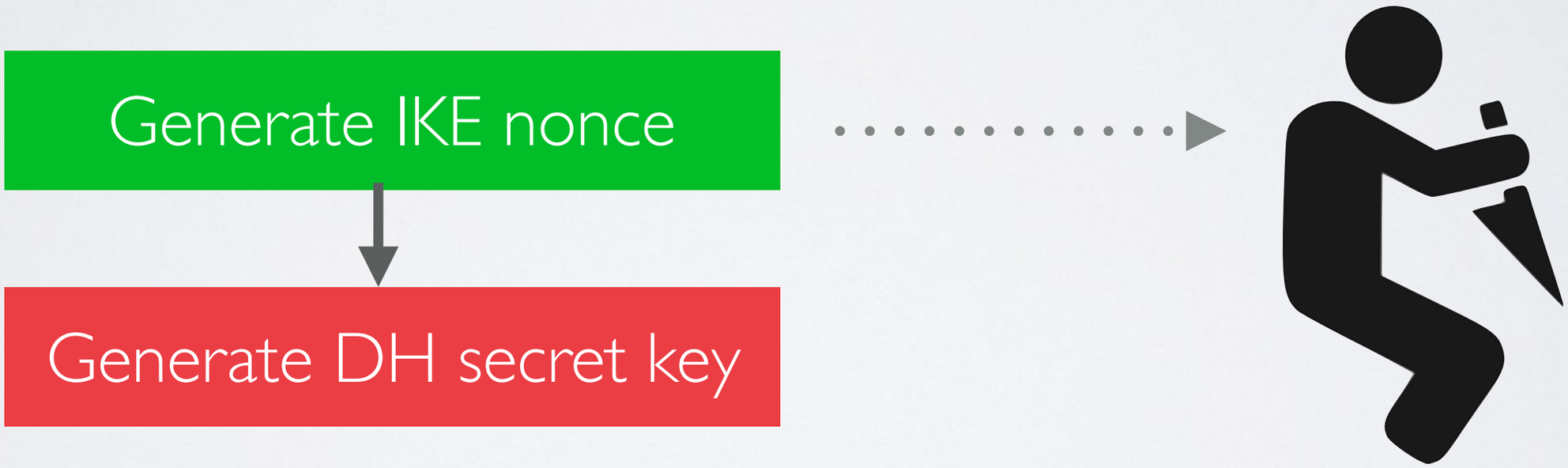
Credit: William Pinckaers

Revised Cascade



Exploiting IKE (Ideal)

- Like many protocols, outputs *nonces*
  - In ScreenOS 6.1 (pre-Dual EC): 20 bytes
    In ScreenOS 6.2 (with Dual EC): 32 bytes
    (>= 28 bytes is sufficient to recover Dual EC state)



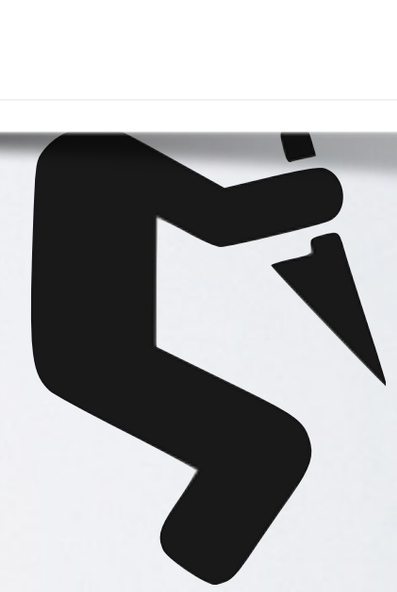
Exploiting IKE (Ideal)

- L
  -

This is (apparently) <u>not</u> what Juniper does

Generate IKE nonce

Generate DH secret key



recompute DH secret key

Exploiting IKE (ScreenOS 6.1)

- All versions of ScreenOS appear to generate key first Nonce second
  - Even with Dual EC, hinders the attack

Generate DH secret key

Generate IKE nonce

(must wait for next handshake)

Exploiting IKE (ScreenOS 6.2)

- ScreenOS 6.2 (the version that adds Dual EC)
  - Adds a *nonce pre-generation queue*
  - Effectively ensures that nonces are always generated first

Generate IKE nonce

Generate DH secret key

recompute DH secret key

Exploitability criteria

JUNIPER NETWORKS JUNIPER NETWORKS

	2007	2008	2012
	ScreenOS 6.1r20	ScreenOS 6.2r1	ScreenOS 6.2r15
Dual EC	X		
32-byte nonces	X		
Nonce generated before Diffie-Hellman	X		
PRNG reseeds each block	X		
Global/ANSI loop	X		
Attacker “Q”	X		

Exploitability criteria

JUNIPER NETWORKS JUNIPER NETWORKS

	2007	2008	2012
	ScreenOS 6.1r20	ScreenOS 6.2r1	ScreenOS 6.2r15
Dual EC	X	✔	
32-byte nonces	X	✔	
Nonce generated before Diffie-Hellman	X	✔	
PRNG reseeds each block	X	✔	
Global/ANSI loop	X	✔	
Attacker “Q”	X	X	

Exploitability criteria

JUNIPER NETWORKS JUNIPER NETWORKS

	2007	2008	2012
	ScreenOS 6.1r20	ScreenOS 6.2r1	ScreenOS 6.2r15
Dual EC	X	✔	✔
32-byte nonces	X	✔	✔
Nonce generated before Diffie-Hellman	X	✔	✔
PRNG reseeds each block	X	✔	✔
Global/ANSI loop	X	✔	✔
Attacker “Q”	X	X	✔

What does it all mean?

- **What we should <u>not</u> take away from this:**
  - *"A bad thing happened to a good company"*

What does it all mean?

- **What we should <u>not</u> take away from this:**
  - *"A bad thing happened to a good company"*
  - Because it's <u>not</u> one company

- **(TS//SI//REL TO USA, FVEY) Reach full operating capability for SIGINT access to a major Internet Peer-to-Peer voice and text communications system.**
- **(TS//SI//REL TO USA, FVEY) Complete enabling for the two leading encryption chips used in Virtual Private Network and Web encryption devices. [CCP_00009]**

(It's not even one PRNG algorithm…)

What does it all mean?

- **What we should <u>not</u> take away from this:**
  - *"The only solution to this is to use safe building blocks"*
  - I.e., it's not the protocol designer's problem, not the system designer's problem

Protocol design *matters*

- **Example: PSK in IKEv1**
  - PSK is fed into the KDF
  - If PSK is high entropy, devices not exploitable!

Protocol design *matters*

- **Example: PSK in IKEv1**
  - PSK is fed into the KDF
  - If PSK is high entropy, devices not exploitable
- **Example: PSK in IKEv2**
  - PSK is not fed into the KDF
  - Devices may be exploitable!

Type 1: IPSec

- IPSec: IP Security
- Complete paired IKE
  - Common UDP ports: 500 and 4500
- Pre-Shared Key (PSK)
  - Router configuration (good source for PSKs)
- Encrypted Payload (ESP or AH)
  - Next Protocol 50 or 51
  - XKEYSCORE Queries
  - Full log DNI search
  - AppID/Fingerprints: “vpn/*”, “vpn/esp”, “vpn/isakmp”, “vpn/ikev2”, “vpn/ikev2_content”

http://www.spiegel.de/media/media-35515.pdf

This should drive our research

- **Mostly it doesn't. But some notable exceptions:**
  - Algorithm Substitution Attacks (Bellare, Paterson, Rogaway)
  - Kleptography (Young, Yung)
  - Formal Treatments of RNGs (Dodis *et al.)*
  - *Formal Verification Approaches (INRIA, MSR, Princeton)*

February 16, 2016

A Message to Our Customers

The Need for Encryption

Smartphones, led by iPhone, have become an essential part of our lives. People use them to store an incredible amount of personal information, from our private conversations to our photos, our music, our notes, our calendars and contacts, our financial information and health data, even where we have been and where we are going.

All that information needs to be protected from hackers and criminals who want to access it, steal it, and use it without our knowledge or permission. Customers expect Apple and other technology companies to do everything in our power to protect their personal information, and at Apple we are deeply committed to safeguarding their data.

Compromising the security of our personal information can ultimately put our personal safety at risk. That is why encryption has become so important to all of us.

For many years, we have used encryption to protect our customers' personal data because we believe it's the only way to keep their information safe. We have even put that data out of our own reach, because we believe the contents of your iPhone are none of our business.

Apple

- Apple is way ahead of the game
- And yet, they're currently flailing because they did not anticipate what they'd be asked to do
  - In particular, there's strong evidence that their Secure Enclave firmware can be flashed in the same way as the older devices

FBI criticises firm for being unable to read iPhone users' messages after Justice Department obtains iMessage ruling

FBI director James Comey describes Apple's encryption as 'a closet that could never be opened'. Photograph: Mark Wilson/Getty