Henri,

Thanks, but:
1- Can we say categorically that the method described does not work anymore and that the patch blocks all attacks of that type?

2- An other example of PGP messed-up file. Any news from them on fixing that?

Thanks

Alan A. Guggenheim
CEO
NagraStar
Alan.Guggenheim@NagraStar.com <mailto:Alan.Guggenheim@NagraStar.com>
Tel: +1 (303) 706-5707

-----Original Message-----
**From:** Henri Kudelski
**Sent:** Tuesday, April 17, 2001 6:14 AM
**To:** Alan Guggenheim; Henri Kudelski
**Cc:** Groux Cédric; Christophe Nicolas; Conus Joël
**Subject:** RE: 4-16-01 Report

*** PGP Signature Status: good
*** Signer: Henri Kudelski
*** Signed: 4/17/2001 6:09:27 AM
*** Verified: 4/17/2001 7:01:26 AM
*** BEGIN PGP DECRYPTED/VERIFIED MESSAGE ***

Hi Alan, The last post gives you the answer: I havent analyze the feb update yet, but if it does check to make sure the packet size is under 64 byte, then there is no way to send a packet to wrap around and overwrite the stack. The buffer is 100 bytes, the overflow code is around 200 bytes if im not mistaken. Any packet 64 bytes or less will fit into the receive buffer with no problems." Regards, Henri ---- Original Message---- From: Alan Guggenheim [mailto:alan.guggenheim@nagrastar.com] Sent: Tuesday, April 17, 2001 02:25 To: Henri Kudelski Cc: Groux Cédric; Christophe Nicolas; Conus Joël Subject: RE: 4-16-01 Report *** PGP Signature Status: good *** Signer: Alan A. Guggenheim *** Signed: 17.4.2001 2:27:37 AM *** Verified: 17.4.2001 1:59:26 PM *** BEGIN PGP DECRYPTED/VERIFIED MESSAGE *** What is the exact situation on the following: " Innermatrix I wonder..... bYrd a reprint from material by Stuntguy..... "Finally, new information released to the public (and which I've verified personally) indicates that there's a bug in the 16CF54 silicon. It seems as though the addressing hardware for the RAM in the card doesn't fully decode the address being requested, and as a result, reads and writes to RAM at addresses above $021F wrap to $0020. So, for example, executing a STA $1FF,X instruction with X=$41 will result not in a useless write to $240 (which isn't a valid RAM address), but rather a perfectly fine write

Henri,

Thanks, but:
1- Can we say categorically that the method described does not work anymore and that the patch blocks all attacks of that type?

2- An other example of PGP messed-up file. Any news from them on fixing that?

Thanks

Alan A. Guggenheim
*CEO*
NagraStar
Alan.Guggenheim@NagraStar.com <mailto:Alan.Guggenheim@NagraStar.com>
Tel: +1 (303) 706-5707

—Original Message—
**From:** Henri Kudelski
**Sent:** Tuesday, April 17, 2001 6:14 AM
**To:** Alan Guggenheim; Henri Kudelski
**Cc:** Groux Cédric; Christophe Nicolas; Conus Joël
**Subject:** RE: 4-16-01 Report

*** PGP Signature Status: good
*** Signer: Henri Kudelski
*** Signed: 4/17/2001 6:09:27 AM
*** Verified: 4/17/2001 7:01:26 AM
*** BEGIN PGP DECRYPTED/VERIFIED MESSAGE ***

Hi Alan, The last post gives you the answer: I havent analyze the feb update yet, but if it does check to make sure the packet size is under 64 byte, then there is no way to send a packet to wrap around and overwrite the stack. The buffer is 100 bytes, the overflow code is around 200 bytes if im not mistaken. Any packet 64 bytes or less will fit into the receive buffer with no problems." Regards, Henri —— Original Message—— From: Alan Guggenheim [mailto:alan.guggenheim@nagrastar.com] Sent: Tuesday, April 17, 2001 02:25 To: Henri Kudelski Cc: Groux Cédric; Christophe Nicolas; Conus Joël Subject: RE: 4-16-01 Report *** PGP Signature Status: good *** Signer: Alan A. Guggenheim *** Signed: 17.4.2001 2:27:37 AM *** Verified: 17.4.2001 1:59:26 PM *** BEGIN PGP DECRYPTED/VERIFIED MESSAGE *** What is the exact situation on the following: " Innermatrix I wonder..... bYrd a reprint from material by Stuntguy..... "Finally, new information released to the public (and which I've verified personally) indicates that there's a bug in the 16CF54 silicon. It seems as though the addressing hardware for the RAM in the card doesn't fully decode the address being requested, and as a result, reads and writes to RAM at addresses above $021F wrap to $0020. So, for example, executing a STA $1FF,X instruction with X=$41 will result not in a useless write to $240 (which isn't a valid RAM address), but rather a perfectly fine write

to $0040. Although this might not seem terribly useful, this in fact allows a very handy backdoor into the EchoStar cards: Due to the fact that the serial I/O buffer is located where it is, it's possible to take advantage of this shortcoming and cause the "receive message" routine to write received serial data into the stack area, where you could then replace the return address that's normally there (to get back to the main idle loop from the interrupt-driven message receive routine) with an address that would cause raw code that you sent as part of your message to be executed. Obviously, this allows lots of interesting things to happen, such as dumping the EEPROM-based password, which would effectively allow unlimited access to the factory backdoors." Quantum Ha haaaa... This would be devastating for them. Did you get this info from a credible source? Darkness Stuntguy! It is in his FAQ. But I think this is what they just closed, isn't it? Any data greater than $64 bytes, which is what you need to wrap, results in a mark being sent to your card for ECM... I am not 100% sure of this, but that is what I have heard... Quantum I have GOT to get that FAQ. No this is not the same buffer-overflow vulnerability that they closed. This one is a flaw in the design of the card microprocessor (ST16CF54A), which cannot be ameliorated and could give access to all the storage areas of the card, and could allow unlooping! This is more of a wrap-around attack, than a buffer-overflow. Devastating for them. Only solution is new cards. Only thing lacking now is the tools & the algo in the cryptoprocessor ASIC. dicknetguy Quantum, Get the FAQ here. http://www.geocities.com/tooliobollocks/ dishcoder This is the overflow that has allowed dumping of rom2 and rom3 cards. By overflowing the receive buffer, we insert code starting at $20 up to $7F, which overwrites the stack which sits from $40 to $7F. If you look at the rom3 overflow packet sent to the card to dump eeprom content, it writes the eeprom dump code at $60 and the stack gets overwritten at the end of the packet which executes the code at $60, allowing dumping of the eeprom (or rom depending on code). The feb update has sucesfully close this hole for rom3 cards. rom2 still open, dont think they have any space in eeprom to fix the bug. Old news really . bYrd "So, for example, executing a STA $1FF,X instruction with X=$41 will result not in a useless write to $240 (which isn't a valid RAM address), but rather a perfectly fine write to $0040." ..... In this example by Stuntguy, is the buffer being overflowed or is this a write to a $0040 that could still be executed on a "locked" card ? Quantum I was waiting for someone to throw cold water on this. A classic buffer overflow is where you send enough data in one packet to overflow the input buffer. Writing then continues into the stack, where you make sure and put in some choice code the processor will get back to, to bend it to your will. Most all software not specifically protected from this is subject to it, as were the cards before the Feb update. The Feb update watches packet size & if it exceeds 64 bytes, writes FF to $E011, permanently marking the card. This mark cannot be undone, no matter what some claim, as it is a PROM. As I am understanding Stuntguy, if you send a packet smaller than the update's limit (under the radar), which contains an invalid address (too high) it will still operate, but will wrap around that address to do the operation somewhere interesting. And no update could tell the difference. I say this is something very useful which will be very hard for them to fix. Where are our gifted coders? Prove me wrong somebody. dishcoder I havent analyze the feb update yet, but if it does check to make sure the packet size is under 64 byte, then there is no way to send a packet to wrap around and overwrite the stack. The buffer is 100 bytes, the overflow code is around 200 bytes if im not mistaken. Any packet 64 bytes or less will fit into the

receive buffer with no problems." Alan A. Guggenheim CEO NagraStar Alan.Guggenheim@NagraStar.com < mailto:Alan.Guggenheim@NagraStar.com > Tel: +1 (303) 706-5707 —— Original Message—— From: jj gee Sent: Monday, April 16, 2001 5:37 PM To: Alan Guggenheim; Cedrix Groux; Christophe Gaillard; Christophe Nicolas; Henri Kudelski; Joel Conus Subject: FW: 4-16-01 Report *** PGP Signature Status: good *** Signer: JJ Gee *** Signed: 4/16/2001 5:37:13 PM *** Verified: 4/16/2001 6:26:40 PM *** BEGIN PGP DECRYPTED/VERIFIED MESSAGE *** ——Original Message—— From: Toy, Donald [mailto:Donald.Toy@echostar.com] Sent: Monday, April 16, 2001 4:45 PM To: Alan Guggenheim; Chris Melton; Dave Kummer; JJ Gee; John Mathews; Kranti Kilaru; Renee Coltharp; Rex Povenmire; Richard Johnson; Russ Densmore; Terry Pattison Subject: 4-16-01 Report *** PGP Signature Status: good *** Signer: Donald Toy (Invalid) *** Signed: 4/16/2001 4:44:32 PM *** Verified: 4/16/2001 5:27:05 PM *** BEGIN PGP DECRYPTED/VERIFIED MESSAGE *** Attached is the Daily Piracy report for 4/16/01. A copy is being sent to Mr. Dugan. Don Toy Echostar Technologies Corp. Dish Network Security - Signal Integrity 303-706-5291 303-706-5723 (fax) mailto:donald.toy@echostar.com *** END PGP DECRYPTED/VERIFIED MESSAGE *** <> *** END PGP DECRYPTED/VERIFIED MESSAGE *** *** END PGP DECRYPTED/VERIFIED MESSAGE ***

*** END PGP DECRYPTED/VERIFIED MESSAGE ***