Spread Spectrum Satcom Hacking: Attacking The GlobalStar Simplex Data Service

Recently there have been several highly publicized talks about satellite hacking. However, most only touch on the theoretical rather than demonstrate actual vulnerabilities and real world attack scenarios. This talk will demystify some of the technologies behind satellite communications and do what no one has done before - take the audience step-by-step from reverse engineering to exploitation of the GlobalStar simplex satcom protocol and demonstrate a full blown signals intelligence collection and spoofing capability. I will also demonstrate how an attacker might simulate critical conditions in satellite connected SCADA systems.

In recent years, Globalstar has gained popularity with the introduction of its consumer focused SPOT asset-tracking solutions. During the session, I'll deconstruct the transmitters used in these (and commercial) solutions and reveal design and implementation flaws that result in the ability to intercept, spoof, falsify, and intelligently jam communications. Due to design tradeoffs these vulnerabilities are realistically unpatchable and put millions of devices, critical infrastructure, emergency services, and high value assets at risk.

I'll also discuss how functionality put in place for personal locator beacons (devices used to seek emergency services in remote locations - used by travelers, adventurers, pilots, and sailors) could be used to induce panic by simulating a large scale disaster and significantly disrupt a core emergency response service.

To begin, I will provide an overview of GlobalStar's service offering, customers, uses, and discuss why strategically, GlobalStar is an ideal target for the motivated attacker or nation state. I'll provide background on spread spectrum satcom systems and dive into specifics of the GlobalStar bent pipe network topology, communication protocols, and point out the inherent weaknesses and attack vectors that make this target so attractive.

In order to go further, we need to understand more about spread spectrum communication. Previously, it remained the sole territory of PhDs but thanks to recent advances in software defined radio (SDR), this realm is now accessible to the everyday hacker for only hundreds of dollars. Unfortunately, information on practical implementation is scarce and open source toolsets are immature or non-existent. This is surprising since spread spectrum communication is the foundation on which most all modern communication is based (Cellular, WiFi, Bluetooth, ZigBee). I'll share my lessons learned, pitfalls, resources, and everything an attendee will need to know to begin hacking satellites or spread spectrum devices.

Specifically we will be talking about direct sequence spectrum (DSSS) modulation, how it works, and it's meaning in the context of the simplex data service. The discussion will include topics such as binary phase shift keying (BPSK), code division multiple access (CDMA), processing gain, and pseudo noise (PN) sequences / generation, as well as go into the nuances of

modulation and demodulation. At this point I'll discuss how to implement DSSS in software easily and practically.

The talk will then describe a reverse engineering process, utilizing a Universal Software Defined Peripheral along with GNURadio and custom toolsets, to perform the requisite signal analysis and determine modulation parameters of black-box systems. Using this methodology, parameters for the simplex data service will be revealed to the audience. With these, I will discuss how to develop a robust receiver and use it to determine the simplex data service on-air packet structure. I will then elaborate on the structure (preamble, device ID, packet number, etc.) including the proprietary 24-bit CRC field and demonstrate how I was able to break the CRC.

With the CRC broken, packet structure known, and spreading parameters deciphered, the audience will be shown how to develop the homebrew transceiver used in this research. With this, I will demonstrate how to intercept tracking data from asset trackers and how this data can be mapped to determine pattern-of-life and the identity of the asset. Next, by crafting custom packets, I will demonstrate how to falsify data and spoof any device on the GlobalStar network.

At this point, it's important to discuss the real world implications for the consumer and corporate customer. I'll cover what information is at risk, how it affects the end user, what a criminal could do with this data, and how certain SCADA networks may be affected. To drive the point home, I will demonstrate a practical real world attack scenario to covertly hijack a high value asset being monitored by the GlobalStar service. This will be achieved by means of disabling the assets tracker and spoofing its location as to not raise suspicion while the hijacking is in progress.

In conclusion I'll address the state of security of legacy satellite systems and what can be done to resolve outstanding security issues. Tips on secure use will be provided for vendors, consumers, hackers, integrators, and first responders alike. I'll drive a few key points home, such as why one should not solely rely on pseudo noise for security as is commonly done, why these systems are so insecure in the first place, and what the industry needs to do going forward. The audience will leave this presentation with an in depth knowledge of satellite communication systems, spread spectrum communications, and the ability to try their hand at hacking their very own satellite. Custom toolsets, schematics, modulation parameters, and resources will be available online at the completion of the session.

Appendix I

DSSS Modulation Parameters	
Frequency	1.61125 Ghz (other frequencies allocated)
Data Modulation	BPSK - Non-differential Encoding
Data Rate	~100 bits per second
Chip Rate	1.25 million chips per second
PN Sequence Type	M-Sequence
PN Sequence Length	255
PN Generating Shift Register Type	GLFSR
Shift Register Polynomial Degree	8
Shift Register Mask	166
Shift Register Seed	59

Actual PN Sequence:

1111111100101101011011101010101110010011011010011001101000111011011000100010011110100100100001111000101001110001111101011110011101000010101100101000101100000110010001100001101111110111000010000010010101001011111000000111001100011010100000000101110111011000

	Simplex Data Packet (144 bits)							
	Preamble (10 bits)	Manufacturer ID (3 bits)	Transmitter ID (23 bits)	Message Number (4 bits)	Packet Number (4 bits)	Sequence Number (4 bits)	User Data (72 bits)	CRC (24 bits)
Sample Data	0000001011	001	01001101100011111010000	0101	0000	0000	010011110000000100000010000010000000000000000100000000000000000000000000000000	000011001000001010010011