Curriculum Vitae

ari.edelkind@episec.com
New York, NY
917 968 6991

Ari Edelkind
July, 2004

Highlights

- I've been working with unix systems for over 9 years now, 7 professionally.
- I have experience with Linux, FreeBSD, BSDI, Solaris, IRIX, HP-UX, and others. A list of more operating systems and greater detail is available in the document *Skill Set and Certifications*. When I refer to 'unix' (lower case), I am including Linux and other unix-like operating systems listed in that document.
- Most commonly requested unix services (along with many obscure ones) are almost second-nature for me. Some of these include Apache, BIND and djbdns, Sendmail and Qmail, and various SQL servers.
- I'm excellent with C programming (10 years), shell scripting (9 years), perl scripting (7 years), and a number of other programming languages. I am also versed in the arts of socket programming and TCP/IP principles. I learned my first programming language at 7 years of age.
- I have experience writing exploits, developing complex processor instruction code for use in exploits, and reverse engineering software during exploitation.
- I consider the unix kernel familiar ground. I have a thorough understanding of many unix internals, and i have both reviewed and written kernel code for a number of operating systems.
- I have excellent references, including my supervisors from three jobs and as many co-workers as you like.

Work Experience

Bloomberg, L.P. 1/02 – present

System and Network Security

- Reviewed code for security vulnerabilities (C, Fortran, ksh, Perl)
- Wrote exploit code for obscure operating systems such as DGUX (using inline i386 and m88k assembly, AT&T format)
- Wrote custom exploits to improve management security awareness
- Wrote documents outlining security problems, potential exploits, and recommended courses of action
- Wrote utilities to assist in auditing processes (C, Perl)
- Wrote custom solaris kernel modules
- Audited Unix-based systems for security vulnerabilities
- Evaluated procedures and wrote documents on potentially more secure alternatives
- Contact for information on cryptographic protocols and public key cryptography applications
- Acted as the chief forensics analyst after security threats
- Reverse engineered suspicious programs (post security threats), in addition to software with suspected security vulnerabilities (primary tools used were `gdb` and `objdump`)
- Assisted networking, security and system administration groups with knowledge of unix-related programs and internals
- Prepared custom seminars on unix and security for teaching to the rest of the group
- Ported software to AIX, DGUX, HP-UX

Operating systems:

AIX 5.2
DGUX 4.20
FreeBSD 4.4–5.1
HP-UX 10.20, 11.23
Linux 2.2, 2.4
Solaris 2.6–8

Starmedia Network, Inc. *(laid off in good standing)* 3/01 – 12/01
Senior System Administrator, security team 9 months

- One of the chief system administrators in charge of 300 unix hosts
- Exclusively responsible for account and privilege management
- Assigned most group programming and scripting projects (C, Perl, PHP, ksh, expect/Tcl)
- Evaluated applications for security issues, and proposed workarounds
- Evaluated applications for deployment (i.e. cryptocard)
- Policy development and deployment
- General system administration tasks, including
  - Server administration and support
  - Application administration and support (Apache, Netscape Enterprise Server, StoryServer, Sendmail, Qmail, Oracle, BIND, Veritas volume manager, EMC Symmetrix utilities)

Operating systems:
FreeBSD 3.1–4.3
Solaris 2.6–8

Taos Mountain, Inc. *(laid off in good standing)* 12/00 – 2/01
System Administrator (consultant) 3 months

- Go on consulting assignments, performing system administration tasks for clients (Unix and NT server administration)
- While not on assignment, learn useful material and take company-sponsored certification courses
  - Solaris SA1 certification
  - Check Point Firewall-1 administration
  - Experimentation with kerberized radius servers
  - Studied HP-UX

Operating systems:
HP-UX 10.20
Linux 2.2
Solaris 7, 8
Windows NT

Public Health Research Institute *(left to work at Taos)* 3/99 – 12/00
Unix System and Network Administrator 21 months

- Network servers consisted of Solaris, IRIX, and Linux hosts (majority was Solaris)
- Networking equipment consisted mainly of Cisco routers and switches
- Upgraded all systems, applied vendor patches to each
- Secured necessary programs and services; wrote security patches for software and submitted them to public mailing lists
- Ported various software to SYSV architectures (Solaris, IRIX)
- General system administration tasks, including
  - User account administration with NIS+ and custom scripts
  - Network backups using custom scripts (sh/ksh, tcl), burt, and amanda
  - File servers, using NFS (with SecureRPC), Netatalk, and Samba
  - Service installation, securing, and administration (Apache, Sendmail, Qmail, BIND, qi/ph (the CSO directory nameserver)
- Wrote scripts and programs to increase security and usability (C, ksh, Perl, PHP, Tcl)
- Provided a secure and productive shell environment for users (mainly scientists and scientific assistants).

Operating systems:
IRIX 6.3 – 6.5.4
Linux 2.2
OpenStep 4.2
Solaris 2.3 – 7
Cisco IOS and Supervisor Software

Also supported:
MacOS 7.6 – 9
Windows 95, 98, 2000, NT

Lamers Net Web and Shell Services *(company folded)* 1/97 – 12/98
Security/System Administrator and Consultant 24 months

- Local and network security to protect against malicious users and external attackers
- Supported the Unix environment for users and other (less knowledgeable) administrators
- Wrote custom scripts and programs (C, Perl, ksh)
- General Unix service administration: web servers (NCSA and Apache), name servers (BIND), mail servers (Sendmail), irc servers (IRCD-Hybrid)

Operating systems:

BSDI 3.0, 3.1

(Freelance) *(left to work at PHRI)* 2/96 – 2/99
Freelance consultant 37 months

- Serviced both individuals and companies
- Tutoring in unix- and windows-based systems
- Network design and installation
- System maintenance and general administration
- Host security work (most requested)

Operating systems:

FreeBSD 2.1 – 3.0
Linux 2.0
Windows 95, 98, NT

Linux General Store *(left to work at PHRI)* 5/98 – 2/99
Volunteer consultant 9 months

- Experimented with new programs and operating systems (i.e. not just linux)
- Wrote custom scripts and programs for store use (C, Perl)
- Assisted users with installation and administration problems
- Participated in the setup of the wireless network and internet access
- Assisted in beowulf cluster assembly from obsolete hardware

Custom Works

I have written a good bit of software over the past years. Some of the more useful packages (written in C) are briefly noted below. For a more complete list of publicly available software and more encompassing descriptions, covering a variety of programming languages, see:

`http://www.episec.com/people/edelkind/`

`lx_lib` Structural Memory Library 11/00 – present
`http://www.episec.com/people/edelkind/lx_lib.html`

`lx_lib` is a structural data library, designed for security, functionality, speed, and convenience, in that order. With `lx_lib`, memory allocation is handled using structures, promoting (among other things) more secure programming practices. This is a fully functional work; additional functionality is added as desired or requested.

Structural `get_opts` 4/01 – present
`http://www.episec.com/people/edelkind/get_opts.html`

Using a slightly different option parsing method than most, `get_opts` parses desired arguments into memory structures, which are easily accessed manipulated. `get_opts` is designed to be an efficient, easily usable, and extensible option parsing library. This is a complete and fully functional work.

Network Authentication Wrapper (`libnaw`) 9/01 – present
`http://www.episec.com/people/edelkind/libnaw.html`

Many programs insufficiently authenticate network connections. Some skip this vital step completely. `libnaw` wraps network library calls, and authenticates based on the configuration file you create for it. Using `libnaw`, you can (once complete) force mutual cryptographic authentication, so that both the client and server can be sure who's really on the other end. `libnaw` is a partially functional work, and is under heavy development.

`slowget` Load Tester 9/01 – 9/01
`http://www.episec.com/people/edelkind/slowget.html`

`slowget` is a metered load testing program allowing site administrators to emulate slow connections to their servers. This is a highly important, yet commonly overlooked, element to generating accurate load testing results. `slowget` is a complete and fully functional work.

`fmtlib` Binary Auditing Tool 07/02 – 07/02

`http://www.episec.com/people/edelkind/fmtlib.html`

Format string misuse is a dangerous flaw among today's programs and services, yet problems are prevalent. `fmtlib` is a dynamically loadable shared object that checks calls sent to format string-based functions and checks to see if the format string was passed using writable memory (i.e. memory that could have been loaded with user input). Instances are logged, and may be reviewed at a later date. The current version works with both FreeBSD and DGUX, and may easily be ported to other operating systems and arcitectures.

`socketwinch` Socket Redirection Tool 09/03 – 09/03

`http://www.episec.com/people/edelkind/socketwinch.html`

`socketwinch`is a socket redirection program that reads from many sockets and writes to one single socket. It is useful for running services that log to unix domain sockets in chroot(2) environments, converting between STREAM and DGRAM styles of sockets, etc.

Papers

Securing Insecure Programs: Circumventing the Designer Bug 2/01
submitted to Dr. Dobbs Journal (available upon request)

Exploit Instruction Code Construction: assisting the manipulation of services on obscure operating systems 7/02
`http://www.episec.com/people/edelkind/papers/shellcode.html`