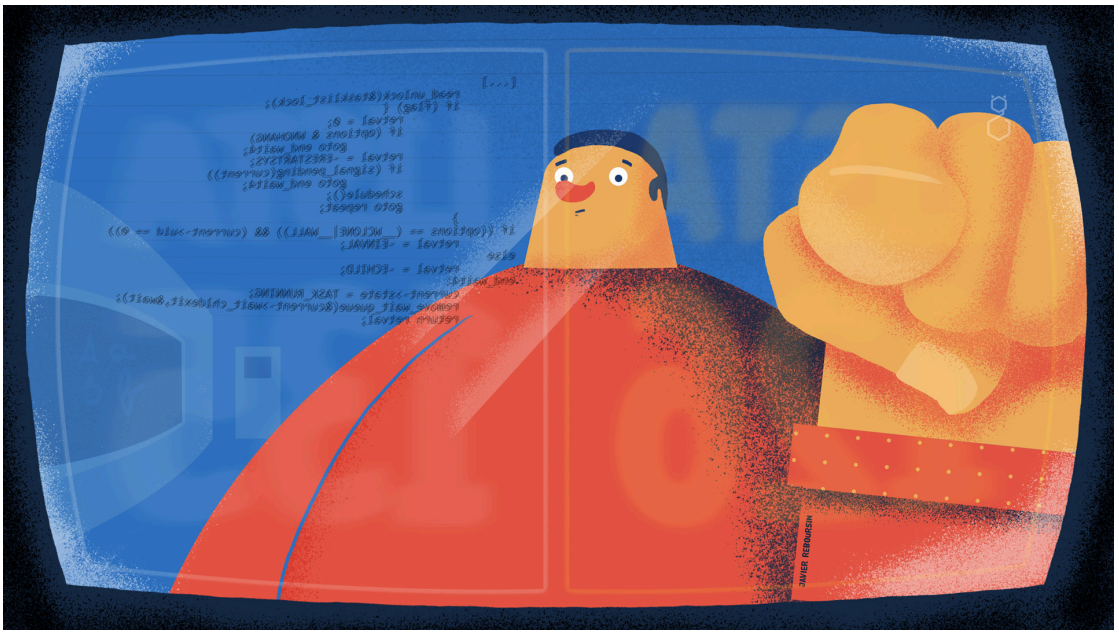

31/10/2016

Vot no

TXT NICOLAS D'IPPOLITO IMG JAVIER REBOURSIN

¿Qué sabemos de los sistemas de voto electrónico? ¿Es seguro? ¿Es auditable?

Hablemos de las elecciones. Hablemos de la boleta única electrónica.
Sin preámbulos ni entrada en calor, **hablemos de la posibilidad de que alguien pueda manipular las máquinas que usaríamos para votar.** Podemos empezar por prestar atención al siguiente fragmento de código:

```
[...]
    read_unlock(&tasklist_lock);
    if (flag) {
        retval = 0;
        if (options & WNOHANG)
```

```
                goto end_wait4;
        retval = -ERESTARTSYS;
        if (signal_pending(current))
                goto end_wait4;
        schedule();
        goto repeat;
    }
if ((options == (__WCLONE|__WALL)) && (current->uid == 0))
    retval = -EINVAL;
else
    retval = -ECHILD;
end_wait4:
    current->state = TASK_RUNNING;
    remove_wait_queue(&current->wait_chldexit,&wait);
    return retval;
}
```

Es importante verlo detenidamente, sé que parece tedioso pero vale la pena el esfuerzo. Acá va de nuevo:

```
[...]
    read_unlock(&tasklist_lock);
    if (flag) {
        retval = 0;
        if (options & WNOHANG)
            goto end_wait4;
        retval = -ERESTARTSYS;
        if (signal_pending(current))
            goto end_wait4;
        schedule();
        goto repeat;
    }
```

```
if ((options == (__WCLONE|__WALL)) && (current->uid = 0))
    retval = -EINVAL;
else
    retval = -ECHILD;
end_wait4:
    current->state = TASK_RUNNING;
    remove_wait_queue(&current->wait_chldexit,&wait);
    return retval;
}
```

¿Qué es lo importante de este código? Que no es uno, son dos distintos, con una pequeña diferencia: uno de ellos tiene un signo '=' de menos. Es la única diferencia, dos miserables rayitas, pero con una implicancia no menor: si el segundo estuviese corriendo en una computadora, ésta podría ser hackeada con facilidad. Se trata de un caso real del año 2003, de código que se encuentra en la parte central del sistema operativo Linux. La diferencia entre la versión correcta y la que te hace sonar es tan sutil que es muy difícil de detectar, incluso por expertos (para ser riguroso, este caso se detectó con facilidad porque se trató de una modificación a un código ya existente y hay herramientas que muestran sólo aquellas líneas que cambiaron, que en este caso eran sólo dos, cuando hay que auditar una pieza de software desde cero no se cuenta con esa ventaja).
Me adelanto a la objeción: *si fuera tan difícil, ¿cómo saben las empresas que venden software que sus productos no tienen fallas?* La respuesta es muy sencilla: **no lo saben**. Y no se trata de que en su ansia desmedida por apropiarse de la renta saquen productos a medio cocinar. Bueno, a veces un poquito sí, pero hay dificultades mucho más de fondo.

Seguro es el que probó y confió

Ahora, podríamos probar con probar, ¿no? Es decir, si uno quiere saber si una pieza de software falla en algún caso, puede probarla y con eso alcanza, ¿no? **No, la verdad es que con probar no alcanza**. El *testing* es la disciplina informática

encargada de probar una pieza de software buscando incrementar la confianza que se tiene en que opera como debe. Funciona así: uno prepara una batería de casos de prueba, que son descripciones paso a paso de qué hacer con el sistema, y compara el resultado obtenido con el esperado. Si no son iguales, acabamos de encontrar un defecto (lo que coloquialmente se llama 'un bug').

Por otro lado, si sí lo son, la única garantía que tenemos es que esa interacción (y no otra, y mucho menos todas) funcionó bien la vez que la probamos, pero tampoco es que estamos tan seguros. Aún si corremos otra vez exactamente la misma secuencia, este segundo intento podría fallar porque no sabemos si el programa en cuestión tiene en cuenta alguna 'variable invisible', como por ejemplo la hora de la computadora, y entonces se comporta de una forma cuando esa variable es una (digamos, a las 9:13 de un martes), y de forma distinta en otra (por ejemplo, a las 4:12 de la madrugada del sábado). El que no se comporte distinto a las 4:12 del sábado, que tire la primera piedra.

Aún si tuviésemos el código y pudiésemos mirar todas las variables involucradas, las alternativas crecen exponencialmente y los caminos posibles a probar son millones de millones. Es decir, el testing es un paso fundamental para asegurar la calidad del software, y cuando encuentra un defecto, hay que arreglarlo; pero **el testing nunca puede asegurar la ausencia de defectos**.

Existen métodos automáticos que también ayudan a aumentar la confianza en que el software funcione como se espera. Algunos son muy buenos, pero tampoco son infalibles. La cosa se complica aún más si estamos haciendo testing de seguridad, ya que en ese caso el comportarse correctamente implica que no sólo haga lo que tiene que hacer, sino que no haga nada 'extra'. Un ejemplo puede ser el caso del acceso a un home banking: no sólo debe dejarme entrar solamente con la contraseña correcta, sino que siempre debe transportarla por la red de manera encriptada y un largo etcétera de requisitos que hacen a la fortaleza y seguridad del sistema. Pero hay más: no sólo debe cumplir con estos requisitos, sino que no debe tener ningún tipo de agujero de seguridad: de esos que aprovechan los virus y los hackers para subvertir un sistema y hacer que sucedan cosas no contempladas. En un sistema informático, es muy, muy difícil encontrar fallas sutiles (a modo de ejemplo, un

'bug' de seguridad en un software de código abierto muy usado estuvo presente 20 años hasta que fue hallado u otro que se detectó hace pocos días y estuvo latente por 11 años y presente en las máquinas con las que se votó en la CABA), y ni hablar de aquellas que son introducidas a propósito con la intención de que no puedan hallarse. Valga como ejemplo el bug introducido intencionalmente en el año 2006 en la especificación de (prepárense que parece chino lo que sigue, pero es algo importante) el generador de números al azar 'Dual_EC_DBRG', que es una parte central de muchos algoritmos criptográficos. Posteriormente, varios fabricantes implementaron el estándar viciado en sus productos y por ende muchísima información sensible que debía ser protegida por mecanismos criptográficos quedaba al desnudo para el autor del bug, que muchos sospechan que se trataba de la NSA. El incidente recién salió a la luz pública en el año 2013.

Algo que sabemos hace mucho en el mundo del software es que **uno no puede tener garantías de que no hay fallas, y a lo que debe apuntar es a tener un muy alto nivel de confianza en que el sistema en cuestión funcione como se espera**.

¿Qué tan grave es que falle el software? Bueno, si falla Tinder, tal vez nuestros genes no se propaguen (quién te dice, terminamos haciendo un bien a la humanidad). Ahora, si falla un marcapasos, uno pensaría que es bastante más grave. Pero, ¿y si el software altera o permite alterar un resultado electoral? Como el marcapasos, pero de escala país. A eso se lo conoce como la *criticidad*, es decir, qué tan graves son las consecuencias de que falle un sistema.

Cuando se trata de software crítico a lo que debe apuntarse es a hacer nuestro mejor esfuerzo para disminuir la chance de que ese software tenga fallas. Cabría preguntarse por qué se usa software en esos casos si no puede garantizarse que sea seguro. La respuesta es simple: porque las otras alternativas que podrían cumplir las mismas funciones o bien no existen o también pueden fallar.

Tener un alto grado de confianza en un sistema tan crítico como el que interviene en una elección requiere de mucho tiempo de trabajo por parte de un grupo de expertos, que utilizará técnicas como inspección ocular, revisión entre pares, *testing*, análisis estático y dinámico de código, *penetration testing* (no relacionado

con Tinder) y un largo etcétera durante un periodo prolongado de tiempo. Los hallazgos de ese trabajo realimentarán el proceso de diseño y programación del sistema, y el proceso de prueba deberá recomenzar. **Pero, ¿qué pasa en el caso de una elección? ¿Es posible que todos estos controles no sean suficientes? Sí.**

Para nosotros que lo miramos por TV

Para entender por qué, imaginemos el próximo proceso electoral de nuestro país: una compra así de grande debe hacerse por licitación, supongamos que se hace mañana, que procede sin dificultades y se evalúa en tiempo récord.

[Pausa para recuperarse de la risa]

El 1 de enero de 2017 la empresa concesionaria termina por completo de desarrollar los sistemas que intervendrán en la elección, los prueba, los analiza y determina que funcionan correctamente. No estamos hablando de desarrollar una app chiquita; se trata de un sistema muy grande, que incluye mucho código desarrollado por la propia empresa, mucho código desarrollado por terceros, e incluso un sistema operativo o parte de él (que puede tener fallas como las que describimos al comienzo del artículo). Todas y cada una de esas partes deben chequearse profundamente porque funcionan de manera encadenada y el resultado final puede alterarse en cualquiera de ellas. En definitiva, **el sistema entero es tan fuerte como la parte más débil de la cadena**.

Ese 1 de enero, ya curada la resaca, expertos de todos los partidos se reúnen y analizan el sistema utilizando todas las técnicas que mencionamos anteriormente. El sistema completo a probar es muy complejo, dado que contiene hardware (lo que se puede patear) y software (lo que sólo se puede putear), así que les toma 6 meses. Menos tiempo, no es realista; más tiempo, se dificulta llegar a agosto con las máquinas repartidas por los más de 3 millones de km cuadrados de nuestro territorio. Entonces se juntan y en un éxtasis de felicidad brindan porque todas las fallas que fueron encontrando se fueron arreglando (lo cual sólo significa que no encontraron más fallas, no que no existan).

La primera pregunta es: ¿cómo saben que todos auditaron el mismo sistema? Eso es fácil de resolver con el software porque uno puede calcular una firma digital del

código del sistema, y si las firmas coinciden es que auditaron el mismo software. ¿Y el hardware? Bueno, no existe tal cosa como la firma digital del hardware, así que realmente **no hay forma de saber que probaron con el mismo hardware**, y eso es importante porque lo que determina qué va a pasar es la combinación de hardware y software. **Y sí, se pueden poner 'virus' por hardware.**

Pero supongamos que decidimos pasar por alto ese 'detalle' y, en un acto de fe ciega, suponemos que todas las computadoras que se van a usar en la elección fueron bendecidas por Santo Tomás de los Pines en persona y por ende suponemos que el hardware simplemente ejecuta el software en forma fiel, sin interferir con su funcionamiento (insisto con esto: **en un acto de fe**). **¿Cómo sabemos que el software que se va a ejecutar es el que fue auditado?** Deberían reunirse todos, todos, todos, frente a otra de esas computadoras bendecidas y compilarlo ahí mismo (*compilar* es el proceso por el que se pasa de un texto escrito en un lenguaje de programación a esa secuencia de ceros y unos llamada *código de máquina*, que es lo único que 'entiende' una computadora realmente). De nuevo, necesitamos otro acto de fe para ignorar el artículo de Ken Thompson, laureado en 1984 con el Premio Turing (aka 'el Nobel de la Computación'), llamado 'Reflections on Trusting Trust', que explica **cómo el propio compilador puede ser saboteado para, a partir de un programa sin problemas, producir código de máquina malicioso**.

Supongamos que también ignoramos eso, así como el trabajo posterior que lo muestra en la práctica. Tenemos nuestro código de máquina compilado delante de todos, que suponemos que no tiene trampas. Calculamos una firma digital de ese código de máquina y se la pasamos a todos nuestros fiscales. Y ojo acá, que cabe recordar que **ya venimos acumulando dos actos de fe, uno por el hardware, otro por el compilador.**

Llega el día de la elección, viene el empleado del correo con una de esas máquinas que por acto(s) de fe suponemos que no tienen problemas. Trae también su CD o pendrive con el código de máquina que es lo que define qué pasará realmente con ella, y cada uno de los fiscales partidarios chequea con su computadora (que tienen, porque la VAN a necesitar, así que asumimos que hay una computadora

para CADA fiscal) si la firma digital de ese CD o pendrive coincide con el que fue compilado delante de todos. Esto es absolutamente indispensable, porque si los fiscales no pueden corroborar individualmente que el software que se instala en cada máquina es el auditado, no sólo existe la posibilidad real de que se instale otro, sino que además **se deja abierta una puerta para que cualquiera disconforme con el resultado lo atribuya a una adulteración y tenga un punto muy fuerte a su favor.**

Todo esto supone además que no hay que hacer ninguna modificación de último momento (como que la justicia autorice algún cambio en las listas o en la forma de presentarlas, algo que es muy usual), porque habría que repetir todo el proceso de nuevo, ya que cambia el código fuente, el código de máquina y la firma digital.

Nada puede malir sal

¿Qué podría pasar si las máquinas de votación estuvieran 'comprometidas'? (estoy resistiendo el chiste del cybercivil y la cyberfiesta). La verdad, de todo.

Recordemos que, en el formato 'boleta electrónica', el ciudadano elige a sus candidatos y la máquina debe grabar su elección de forma digital y además imprimirlo en formato legible. **Una máquina comprometida o adulterada podría imprimir al candidato A en letras y grabar digitalmente al B**.

No tiene que hacerlo siempre, que sería muy obvio, **puede hacerlo en una cantidad estadísticamente pequeña de casos, lo suficiente como para asignarle una banca de más o de menos a algún partido, o definir un ballotage muy parejo** para una presidencia (pongamosle un 51 a 49 *hipotético*, o recordemos también el referendum en Colombia donde el No acaba de ganar con 50,2% de los votos).

Si de variaciones estadísticas se trata, también podría pasar que el orden al azar en el que aparecen los candidatos no sea tan al azar, dándole prevalencia a alguno. No vamos a hacer un listado exhaustivo, pero analicemos un poco más.

¿Se te ocurre alguna razón para dudar de la Boleta Única Electrónica? Un par.

Unos investigadores independientes reportaron un defecto en el sistema usado en la CABA para las elecciones para Jefe de Gobierno de 2015: permitía cargar varios votos a la vez, algo que ninguna de las auditorías oficiales había notado. Otro investigador descubrió un manejo poco seguro del mecanismo de encripción utilizado, lo que permitía que cualquiera mandara al centro de cómputos resultados como si fuesen oficiales. Lo reportó antes de las elecciones y por supuesto que fue automáticamente respetado y tratado con cuidado. O no: fue allanado y enfrentó un proceso judicial que duró casi un año (así como al pasar, durante ese proceso se determinó que los servidores de la empresa que brindó el servicio habían sido hackeados), con altos costos, hasta que finalmente la justicia determinó que no había cometido ningún delito (y hasta que había dado una genuina mano identificando los problemas). Porque si hay algo que querés cuando reportás un bug en un sistema público crítico es que te traten como un peligroso delincuente y te secuestren todos los aparatos electrónicos, incluyendo compu, laptop, Kindle, y una licuadora que parece que miraba fijo a uno de los gendarmes.

Dilema 5: *Un hacker encuentra una vulnerabilidad en un sistema digital y decide reportarlo...*

Pero, si es electrónico, tiene que ser fantástico

Una objeción que se escucha con frecuencia es que está previsto el escrutinio manual. Analicemos esta posibilidad basándonos en los datos duros del informe final de la Defensoría del Pueblo de la CABA sobre la elección para Jefe de Gobierno de 2015. Según este informe, 'una vez cerrada la mesa, el 83,9% de los presidentes pudo realizar el escrutinio sin inconvenientes. Durante el conteo de votos, sólo el 10,1% de las mesas contó con fiscales que realizaron algún reclamo'. Esto significa que hubo cerca de 730 mesas con reclamos. A 300 votantes por mesa, hay unos 219000 votos en cuestión, muy por encima de los 54000 que definieron la elección en CABA y peligrosamente cerca de los 300000 votos de diferencia que definieron el ballotage presidencial de ese mismo año. De ese informe surge también que un 26,2% de los votantes dijo no haber verificado que el voto impreso coincidiera con lo que había elegido.

Pero además, aún en el caso en que todas las mesas electorales corroboraran el escrutinio electrónico con uno manual, el manual es sólo corroboración de una

planilla que se graba digitalmente en otra boleta electrónica. De nuevo, un software malicioso podría hacer que la grabación tenga cifras adulteradas incluso cuando la propia máquina las siga mostrando como correctas. O tal vez la manipulación podría hacerla la máquina que lee la tarjeta y manda la información a través de Internet hacia el centro de cómputos (que a su vez podría tener software adulterado o hackeado como el de CABA en 2015). No sé cómo vienen ustedes, pero a esta altura ya perdí la cuenta sobre la cantidad de saltos de fe.

Memoria y 'países serios'

El pueblo argentino se ganó el voto universal, secreto y obligatorio en cuotas. Primero nos ganamos el voto (de entrada solamente los varoncitos, aunque ellas conquistaron la universalidad un par de cuotas más tarde), varias dictaduras nos lo sacaron y hubo que reconquistarlo. En el medio de esas peleas, conquistamos el voto secreto, y lo consagramos en la Ley Sáenz Peña.

Entonces teníamos la posibilidad de que nadie supiera a quién votaste, para que no pudieran chantajearte, presionarte, comprarte o vengarse de vos si no les gustaba tu decisión. Esto se manifiesta de manera muy clara cuando tenemos la posibilidad de poner nuestro voto en un sobre idéntico a todos los sobres para después abrir la urna y contar (y sí, hay maneras de manipular y de romper el secreto de voto, pero son fácilmente identificables y auditables por ciudadanos comunes).

Hay que tener memoria, algo que las computadoras también tienen. Justamente el tema de la memoria es central en el argumento de la **Corte Suprema de Alemania que, en el año 2009, prohibió el uso de urnas electrónicas porque contradice el principio de que todos los pasos de la elección estén sometidos al escrutinio público sin requerir conocimientos técnicos especiales**.

Si pudiera elegir un sólo párrafo para ser recordado de todo este texto (que intenta ser exhaustivo respecto de las múltiples aristas a considerar en la adopción o no del voto electrónico y sus variantes), sería éste: **si dependemos de un proceso técnicamente inaccesible para la enorme mayoría de nosotros (salvo los expertos en desarrollo de sistemas de votación electrónica), la transparencia del sistema para el ciudadano común desaparece.**

Con el voto electrónico y sus variantes, a la democracia la vemos pasar, la miramos por TV. Nos cuentan y tenemos que creer o reventar. El pilar de nuestra construcción democrática, la elección, se transforma en algo que no entendemos, que no podemos auditar. No es un detalle menor que el voto sea secreto. Es esencial y nadie nos lo regaló, hubo que luchar mucho para conseguirlo. Con el voto electrónico y sus variantes, puede dejar de serlo. Lo que es peor aún, no sabemos si es o no secreto, y sembrar esa duda (que se vuelve razonable porque el sistema es tan opaco que no hay forma de saber la verdad), alcanza para que alguien pueda manipular nuestra decisión. **El voto no solamente tiene que SER secreto, sino que tiene que LUCIR secreto, para poder ser ejercido sin presiones.**

De imprentas e impresoras

El sistema actual no es perfecto: es cierto que es problemático y costoso distribuir las boletas a todos los cuartos oscuros, y que sería muy bienvenida una alternativa superadora a semejante desafío logístico, especialmente para los partidos chicos. Pero superadora de verdad, no sólo aparentemente. Si vamos a informatizar, pensemos en lo que pasa después de votar, desde hacer un conteo asistido de los votos hechos en papel a cosas mínimas como disponer de un procesador de texto y una impresora en los cuartos oscuros para que las actas no sean manuscritas y haya menos errores de transcripción.

Muchas veces se revolea el argumento de que las máquinas de votación son solamente impresoras. Es un argumento casi gracioso porque las impresoras de hoy en día son solamente otro tipo de computadoras y, como tales, también tienen memoria. Y pueden usar esa memoria para registrar que, por ejemplo, el primer votante votó por A, el segundo por B, el tercero por A de nuevo, y así siguiendo. Con el simple expediente de ir contando, todos los fiscales partidarios pueden saber quién votó primero, quién segundo, etc. No sólo los fiscales, basta con poner a un chabón a fumar en la puerta del cuarto oscuro. **Es decir, no alcanza con que el sistema no manipule los resultados, también hay que garantizar que no registre información de más.**

Análisis de nuestro corresponsal Gato Asia

Y la verdad es que en este caso hace falta poca memoria, o casi ninguna: en cada cuarto oscuro votan unas 300 personas; ese número se codifica con sólo 9 bits.

Si no te resulta obvio que el número 300 se codifica con 9 bits, es un claro ejemplo de como el sistema que estamos discutiendo también te dejó afuera a vos, una persona problablemente educada, curiosa e informada, pero que no tiene conocimientos específicos sobre computación. Qué loco pensar que con esas mismas condiciones sí podrías auditar todo el proceso de voto en papel: saber leer, ser curioso y educarte respecto en el proceso de fiscalización (algo que demora minutos).

Decíamos que alcanza con 9 bits, 9 puntitos escondidos en cualquier parte de la boleta en papel para saber a quién votó cada persona. Si alguien va contando en qué orden vota cada uno de los electores, luego, cuando recuperan las boletas en papel, se miran esos 9 puntitos mínimos, escondidos con algo de cautela, tal vez en el borde de una letra, tal vez simulando ser una mancha de tinta, y se puede reconstruir a quién votó cada elector. ¡En tu cara, Sáenz Peña!

Comparando

El sistema electoral actual no es perfecto y tiene mucho para mejorar. Pero es mucho mejor de lo que se nos quiere hacer creer muchas veces. Si una fuerza política quiere gobernar una ciudad debe concitar las voluntades de la mayor parte de los electores de la ciudad, pero el requisito previo es que tenga un núcleo de personas con un nivel mayor de adhesión, realmente entusiasmados por la propuesta, que estén dispuestos a ser fiscales durante un día. ¿Con qué requisitos? Los básicos: prestar atención, saber leer, sumar y restar, condiciones que en líneas generales cualquier adulto puede cumplir. Sería razonable que las fuerzas políticas que tienen una ambición mayor, como la de gobernar una provincia, tuvieran una cantidad de entusiastas proporcional al tamaño de la provincia. Si no, es muy difícil pensar que la van a poder gobernar. El mismo razonamiento cabe si quieren gobernar un país. Por supuesto que no es fácil, pero gobernar tampoco lo es. Si no podés resolver el problema de conseguir 20 fiscales para ser intendente de tu ciudad, difícilmente puedas resolver los problemas que conlleva la propia intendencia, donde son muchas más las voluntades que deben alinearse, durante mucho más tiempo que un par de domingos cada dos años. Lo mismo si querés gobernar un país.

Por otra parte, es poco verosímil y hasta peligroso que una fuerza política acepte dejar la máquina de votación sin supervisión, con lo que **la implementación de mecanismos electrónicos tampoco elimina la necesidad de fiscales**.

Para los fiscales, el sistema actual podría mejorarse en varios puntos, pero el hecho de poder entrar por la web y ver si el telegrama escaneado tiene tu firma y si la planilla electrónica coincide con lo que está escrito a mano y con tu copia del acta es un punto de control muy fuerte.

Dada la propuesta actual de voto electrónico, con tener fiscales no alcanza, porque en definitiva **los puntos de control establecidos de nada sirven si se pierde el secreto del voto, si lo que se graba en la boleta no refleja la voluntad del elector en todos los casos, o si luego esa información es nuevamente volcada**

a otra computadora que puede manipularla en el proceso.

No es menor decir que este análisis no parte de ser un romántico de los viejos tiempos o un férreo opositor a la ciencia y la tecnología. Lejos estoy de serles fóbico o de no comprenderlas. Es más bien lo contrario en este caso: **entender cómo funciona la tecnología digital nos da herramientas para poder mirarla con ojos críticos.** Justamente por eso entendemos sus limitaciones, como lo hacen casi todos los países desarrollados (para nada tecnofóbicos), que **siguen votando en papel.**

De hecho, si lo que se busca es boleta única, existe la boleta única en papel: en un mismo cacho de árbol tenés a todos los candidatos de todos los partidos y le ponés una cruz a los que prefieras (la única dificultad es que hay que explicarles a los adolescentes que se elige sólo con una cruz y no vale usar otros emoticones).

Decía más arriba que con la boleta electrónica la transparencia se pierde, y es importante recalcar que no reaparece si, en lugar de implementarse a las apuradas, se hiciese con tiempo suficiente.

El sistema está intrínsecamente viciado porque el piso mínimo necesario para entender el proceso electoral electrónico, auditarlo y participar de su control, se vuelve prácticamente inalcanzable. Pasa de requerir habilidades

que se adquieren en la escolaridad básica a volverse una discusión de expertos, cerrada, críptica, y por ende, excluyente.

Somos los ciudadanos y ciudadanas comunes, los que armamos ese nosotros bien grande que trasciende lo que nos aúna y lo que nos separa, los que queremos poder votar de forma secreta y segura, y que nuestro voto se escuche. Que se escuche cristalino, sin intermediarios, dudas o mugre.

Que se escuche exactamente como lo manifestamos, aún cuando el resultado no nos guste, pero sabiendo que genuinamente nos representa.

BONUS TRACK

Esta mañana Nicolas habló con Juan Pablo Varsky sobre la nota, vale la pena escucharlo:

Nota: Para entender un poco más sobre en qué países del mundo se utiliza voto electrónico (spoiler alert: muuuuy pocos) y las controversias que esto encierra, recomendamos leer este artículo de Javier Pallero.

Más refes:

Otra referencia a 'virus' por hardware, en castellano: https://ekoparty.blogspot.com.ar/2014/10/deep-submicron-cpu-backdoors-alfredo.html

Bonus track 2

La opinión de Julian Assagne sobre voto electrónico, aunque debe ser tomada con pinzas, ¿qué sabe él de un tema así?

elgatoylacaja.com/vot-no